



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«МИРЭА – Российский технологический университет»
РТУ МИРЭА

Институт информационных технологий (ИТ)
Кафедра инструментального и прикладного программного обеспечения (ИиППО)

ЗАДАНИЕ
на выполнение курсовой работы

по дисциплине: Проектирование и разработка клиентских частей интернет-ресурсов
по профилю: Разработка и дизайн компьютерных игр и мультимедийных приложений
направления профессиональной подготовки: Программная инженерия (09.03.04)

Студент: Яскевич Матвей Дмитриевич

Группа: ИКБО-21-23

Срок представления к защите: с 18.11.2024 по 09.12.2024

Руководитель: Матчин Василий Тимофеевич, старший преподаватель

Тема: Клиентская часть интернет-ресурса «Электронная зачетная книжка»

Исходные данные: используемые технологии: HTML5, CSS3, JavaScript, редактор кода Visual Studio Code, наличие: интерактивного поведения веб-страниц, межстраничной навигации, внешнего вида страниц, соответствующего современным стандартам веб-разработки; технологий адаптивной верстки для полноценного отображения контента на различных браузерах и видах устройств. Нормативный документ: инструкция по организации и проведению курсового проектирования СМКО МИРЭА 7.5.1/04.И.05-18.

Перечень вопросов, подлежащих разработке, и обязательного графического материала: 1. Провести анализ предметной области разрабатываемой клиентской части интернет-ресурса. 2. Обосновать выбор технологий разработки клиентской части интернет-ресурса. 3. Создать пять и более веб-страниц интернет-ресурса. 4. Организовать межстраничную навигацию. 5. Реализовать слой клиентской логики веб-страниц с применением технологии JavaScript. 6. Провести оптимизацию веб-страниц и размещаемого контента для браузеров и различных видов устройств. 7. Создать презентацию по выполненной курсовой работе.

Руководителем проведен инструктаж по технике безопасности, противопожарной технике и правилам внутреннего распорядка.

Зав. кафедрой ИиППО: [подпись] /Р.Г. Болбаков/, «16» 09 2024 г.

Задание на КР выдал: [подпись] /В.Т. Матчин/, «16» 09 2024 г.

Задание на КР получил: [подпись] /М.Д. Яскевич/, «16» 09 2024 г.

РЕФЕРАТ

Отчет 22 с., 12 рис., 2 источн.
ВЕБ-ПРИЛОЖЕНИЕ, ЗАЧЕТНАЯ КНИЖКА, РАСПИСАНИЕ,
АДАПТИВНЫЙ ДИЗАЙН, ИНТЕРФЕЙС.

Объектом исследования является веб-приложение, предназначенное для автоматизации учебных процессов студентов. Основной целью работы было создание удобного и интуитивно понятного ресурса для управления учебной информацией, включая расписание, оценки и личный профиль пользователя.

В рамках работы были использованы методы веб-разработки, включающие применение современных технологий HTML5, CSS3 и JavaScript. Основная логика приложения сосредоточена на динамическом обновлении данных без необходимости перезагрузки страницы, что обеспечивается использованием клиентских скриптов.

Результатом работы является разработанное приложение, состоящее из пяти функциональных модулей: главной страницы с таймером обратного отсчета и мини-игрой, календаря расписания, страницы с оценками, формы записи на пересдачу и профиля пользователя.

Область применения разработанного веб-приложения – образовательные учреждения, заинтересованные в автоматизации и упрощении взаимодействия студентов с учебной информацией.

Рекомендации по внедрению включают тестирование приложения в учебной среде, настройку конфигурационных данных под конкретное учебное заведение и дальнейшее расширение функционала, включая интеграцию с внешними образовательными системами.

Экономическая эффективность работы заключается в снижении временных затрат студентов и преподавателей на организационные вопросы, а также в улучшении качества взаимодействия в учебном процессе.

СОДЕРЖАНИЕ

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ.....	5
ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ.....	6
ВВЕДЕНИЕ.....	7
1 ОБЩИЕ СВЕДЕНИЯ	8
1.1 Наименование клиентской части интернет-ресурса	8
1.2 Функциональное назначение.....	8
1.3 Программное обеспечение, используемое для разработки и тестирования.....	8
2 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ	9
2.1 Анализ предметной области	9
2.2 Выбор технологий	10
2.3 Создание веб-страниц клиентской части интернет-ресурса с использованием технологий HTML5, CSS3 и JavaScript	10
2.4 Создание межстраничной навигации	12
2.5 Реализация слоя клиенткой логики веб-страниц с применением технологии JavaScript.....	13
3 ОПТИМИЗАЦИЯ КЛИЕНТСКОЙ ЧАСТИ ИНТЕРНЕТ-РЕСУРСА.....	18
3.1 Стилизация приложения	18
3.2 Оптимизация клиентской части интернет-ресурса для стационарных и мобильных устройств	20
ЗАКЛЮЧЕНИЕ	23
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	24

ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В настоящем отчете применяют следующие термины и соответствующие им определения.

Веб-приложение – программное обеспечение, работающее в браузере пользователя, которое использует веб-технологии для выполнения задач, предоставляя интерактивный интерфейс.

Клиентская логика – программная логика, выполняемая на стороне пользователя, предназначенная для обработки данных и взаимодействия с интерфейсом веб-приложения.

Мини-игра – небольшая развлекательная программа, интегрированная в веб-приложение для повышения вовлеченности пользователей.

Конфигурационный файл – файл, содержащий структурированные данные, используемые для настройки и управления параметрами работы приложения.

Адаптивный дизайн – метод разработки интерфейсов, обеспечивающий корректное отображение и функциональность на устройствах с различными размерами экранов.

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

В настоящем отчете применяют следующие сокращения и обозначения.

HTML	–	HyperText Markup Language
CSS	–	Cascading Style Sheets
JS	–	Java Script

ВВЕДЕНИЕ

Электронная зачетная книжка — это современный инструмент для автоматизации и упрощения взаимодействия студентов и образовательных учреждений. Она позволяет хранить и управлять информацией о дисциплинах, успеваемости и расписании в удобном цифровом формате. Клиентская часть такого ресурса играет важную роль в обеспечении удобства и интуитивности взаимодействия пользователя с системой, что напрямую влияет на эффективность использования электронного ресурса.

Целью данной курсовой работы является разработка и тестирование клиентской части интернет-ресурса на тему «Электронная зачетная книжка». Для достижения этой цели были применены современные технологии разработки веб-приложений, такие как HTML, CSS и JavaScript, что позволило создать функциональный, удобный и эстетически привлекательный интерфейс. Разработанный ресурс предоставляет пользователям возможность просматривать информацию о дисциплинах, результатах экзаменов, а также получать доступ к расписанию и персональным данным.

В соответствии с поставленной целью в рамках работы решаются следующие задачи:

- анализ требований и предметной области,
- изучение технологий для разработки клиентской части веб-приложений,
- создание веб-приложения с учетом специфики образовательной среды,
- тестирование и отладка разработанного ресурса.

В процессе выполнения курсовой работы использовались методы анализа технической документации, проектирования интерфейсов и практические навыки разработки клиентской части интернет-ресурсов. Работа основывается на знаниях, полученных в ходе изучения дисциплины «Проектирование и разработка клиентских частей интернет-ресурсов».

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Наименование клиентской части интернет-ресурса

Темой разработанной клиентской части интернет-ресурса является «Клиентская часть интернет-ресурса «Электронная зачетная книжка»».

Название продукта курсовой работы выбрано в соответствии с заданной темой. Разработанное приложение получило название — «Клиентская часть интернет-ресурса «Электронная зачетная книжка»».

1.2 Функциональное назначение

Веб-приложение предназначено для того, чтобы облегчить студентам доступ к их зачетной книжке в электронном формате. Оно предоставляет удобный интерфейс, который позволяет быстро находить информацию о результатах учебы, расписании сессий и изучаемых дисциплинах.

Основная цель разработки — создать понятный и удобный инструмент, который поможет студентам легко ориентироваться в своей академической информации и контролировать успеваемость.

В дальнейшем приложение может быть дополнено новыми функциями и интегрировано с другими образовательными системами. Сейчас оно предоставляет возможность ознакомиться с расписанием, результатами зачетов и экзаменов, а также общей информацией об учебной деятельности.

1.3 Программное обеспечение, используемое для разработки и тестирования

Для работы над проектом использовался редактор WebStorm, который обладает всеми необходимыми инструментами для удобной разработки веб-приложений. Среди его возможностей — подсветка кода, поддержка расширений и интеграция с системами управления проектами.

Тестирование приложения выполнялось в браузерах «Яндекс Браузер» и «Google Chrome», что позволило проверить корректное отображение и работу интерфейса на разных устройствах и платформах.

2 ОПИСАНИЕ ЛОГИЧЕСКОЙ СТРУКТУРЫ

2.1 Анализ предметной области

В рамках разработки клиентской части интернет-ресурса «Электронная зачетная книжка» был проведен анализ предметной области, чтобы определить основные требования к функционалу и интерфейсу приложения.

Цель анализа заключалась в изучении существующих аналогов и выявлении ключевых аспектов, необходимых для удобного взаимодействия студентов с учебной информацией. Были рассмотрены следующие характеристики:

- удобство использования, включая интуитивно понятную навигацию и легкость доступа к нужным данным,
- адаптивность интерфейса, позволяющая работать на различных устройствах, таких как компьютеры, планшеты и смартфоны,
- функциональность, предполагающая наличие необходимых возможностей для работы с расписанием, оценками и другой академической информацией,
- эстетическая привлекательность, влияющая на восприятие и комфорт работы с ресурсом.

В качестве базы для анализа рассматривались интерфейсы образовательных систем, таких как электронные дневники и платформы управления учебным процессом. Результаты исследования показали, что успешные проекты ориентируются на минимализм, четкость интерфейса и возможность быстрого доступа к нужной информации.

Основными требованиями для разработки клиентской части стали:

- реализация удобного отображения дисциплин и оценок,
- создание раздела с расписанием и сведениями о сессии,
- возможность адаптивного отображения данных на устройствах с разными экранами,
- поддержка базовой системы авторизации студентов.

2.2 Выбор технологий

Анализ предметной области помог сформировать список инструментов, которые позволят реализовать поставленные задачи максимально эффективно. Для разработки клиентской части интернет-ресурса «Электронная зачетная книжка» были выбраны следующие технологии:

- HTML5 для разметки страниц и структуры контента,
- CSS3 для стилизации интерфейса и обеспечения адаптивного дизайна,
- JavaScript для реализации интерактивных элементов и функциональности,
- WebStorm в качестве среды разработки благодаря ее удобству, подсветке синтаксиса и поддержке плагинов.

Для проверки работоспособности приложения и тестирования интерфейса использовались браузеры «Яндекс Браузер» и «Google Chrome», обеспечивающие корректное отображение страниц и их взаимодействие с пользователем.

Выбор этих инструментов обоснован их широким использованием в веб-разработке, а также возможностью создания современного и удобного интерфейса, отвечающего всем требованиям проекта.

2.3 Создание веб-страниц клиентской части интернет-ресурса с использованием технологий HTML5, CSS3 и JavaScript

На языке разметки HTML5 были реализованы 5 страниц: главная (разметка представлена на рисунке 1), расписание сессии, оценки по предметам, форма записи на пересдачу и личный профиль пользователя. Дисциплины, персональная информация, содержимое зачетной книжки и прочие данные подгружаются на страницы динамически из конфигурационного JavaScript-файла.

На главной странице реализованы:

- обратный отсчет до ближайшего события, который анализирует список предстоящих зачетов, экзаменов и курсовых работ. На основе текущей даты и времени приложение вычисляет ближайшее событие и отображает время до его начала в формате «дни:часы:минуты:секунды», обновляемое каждую секунду. Если событие началось, таймер заменяется текстом «Событие уже началось!»;

- интерактивная мини-игра, в которой на игровом поле появляются элементы, представляющие дисциплины из текущей сессии. Каждый элемент отображает название предмета и тип контроля (зачет, экзамен). Пользователь набирает очки, кликая по этим элементам, а итоговый результат фиксируется после завершения игры, которая длится 30 секунд.

Страница расписания сессии представлена в виде календаря, который охватывает весь период текущей сессии. Даты с запланированными событиями выделяются визуально. Под каждой датой отображается список мероприятий с подробной информацией о дисциплине, типе контроля и времени проведения. Типы контроля (зачет, экзамен, курсовая работа) выделяются разными цветами для удобства восприятия.

На странице отображается список дисциплин за выбранный семестр. Для каждой дисциплины указываются название, имя преподавателя, тип контроля, дата проведения и оценка. Данные подгружаются из конфигурационного файла config.js. Средний балл за семестр рассчитывается автоматически и отображается с цветовым обозначением:

- зеленый (высокий балл, 4.5 и выше),
- желтый (средний балл, 4.0–4.4),
- красный (низкий балл, ниже 4.0).

В верхней части страницы предусмотрена выпадающая панель для выбора семестра, данные по которому будут отображены.

Форма записи на пересдачу автоматически подгружает список дисциплин с неудовлетворительными оценками (незачет, неудовлетворительно). Пользователь выбирает дисциплину из выпадающего

списка и отправляет заявку на пересдачу. Перед отправкой данных осуществляется проверка корректности заполнения формы.

На странице профиля отображается информация о студенте: имя, группа, возраст, электронная почта, личный номер и другие данные. Также представлена образовательная информация: направление подготовки, срок обучения, формирующее и выпускающее подразделения, тип обучения и прочие параметры. Все данные подгружаются из конфигурационного файла config.js.

```
<main class="content__main">
  <div class="block block_timer">
    <h2 class="block__title">Ближайшее событие</h2>
    <p class="block__description">До события <span class="timer__subject">...</span> осталось:</p>
    <div class="timer">
      <span class="timer__part timer__days">0</span> дней
      <span class="timer__part timer__hours">0</span> часов
      <span class="timer__part timer__minutes">0</span> минут
      <span class="timer__part timer__seconds">0</span> секунд
    </div>
  </div>
  <div class="block block_game">
    <h2 class="block__title">Мини-игра: Попади в зачет или экзамен!</h2>
    <p class="block__description">Кликай на блоки с зачетами и экзаменами, чтобы набрать очки.</p>
    <div class="game">
      <div class="game__field"></div>
      <div class="game__score">0очки: <span class="game__score-value">0</span></div>
    </div>
    <button class="game__start-button">Начать игру</button>
  </div>
</main>
```

Рисунок 1 – Разметка основной части главной страницы

2.4 Создание межстраничной навигации

Для навигации по сайту было создано меню, которое расположено в шапке сайта. С его помощью можно перейти с одной страницы на любую другую. Разметка хедера представлена на рисунке 2.

```

<header class="content__header">
  <div class="content__block block block_header header">
    <div class="header__top top">
      <h2 class="top__site-name">Зачетная книжка</h2>
      <div class="top__user">
        <span class="top__avatar">МБ</span>
        <span class="top__user-name">Матвей</span>
      </div>
    </div>
    <div class="header__menu menu">
      <ul class="menu__list">
        <li class="menu__item"><a class="menu__link" href="index.html">Главная</a></li>
        <li class="menu__item"><a class="menu__link" href="schedule.html">Расписание сессии</a></li>
        <li class="menu__item"><a class="menu__link" href="subjects.html">Зачётная книжка</a></li>
        <li class="menu__item"><a class="menu__link" href="retake.html">Пересдача</a></li>
      </ul>
      <ul class="menu__list">
        <li class="menu__item"><a class="menu__link" href="profile.html">Обо мне</a></li>
      </ul>
    </div>
  </div>
</header>

```

Рисунок 2 – Разметка хедера сайта

Важной особенностью навигации является подсветка активной страницы, которая реализована с использованием JavaScript. Это помогает пользователю визуально определить текущий раздел. Интерфейс хедера показан на рисунке 3.

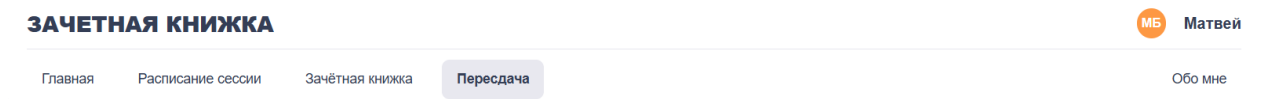


Рисунок 3 – Внешний вид хедера

2.5 Реализация слоя клиенткой логики веб-страниц с применением технологии JavaScript

Клиентская логика веб-страниц была реализована с использованием JavaScript, чтобы обеспечить интерактивность и динамическую работу приложения. Весь функционал распределен по нескольким скриптам, каждый из которых выполняет определенные задачи. Логика организована таким образом, чтобы быть модульной, легко поддерживаемой и расширяемой.

Скрипт index.js отвечает за функциональность главной страницы, включая обратный отсчет до ближайшего события (рисунок 6). Логика реализована следующим образом:

- данные о предстоящих событиях подгружаются из конфигурационного файла `config.js`. Каждое событие преобразуется в объект с датой и временем,

- определяется ближайшее событие, основываясь на текущей дате. Если такие события существуют, таймер отображает оставшееся время до начала, обновляя данные каждую секунду.,

- если события отсутствуют или они уже начались, отображается соответствующее уведомление. Структура скрипта разделена на функции для чтения данных, вычисления времени и обновления интерфейса.

Скрипт `config.js` содержит данные, необходимые для работы приложения: информацию о пользователе, расписание дисциплин, оценки и другие параметры. Данные организованы в виде вложенных объектов, что упрощает их обработку. Например, разделы расписания сгруппированы по семестрам, а параметры пользователя вынесены в отдельную структуру. Часть скрипта представлена на рисунке 4.

Скрипт `game.js` Этот скрипт реализует мини-игру на главной странице. Он генерирует игровое поле с динамически создаваемыми элементами. Каждый элемент представляет дисциплину, данные о которой берутся из текущего расписания. Логика игры заключается в следующем:

- позиции элементов на игровом поле определяются случайным образом,

- пользователь набирает очки, кликая по элементам до их исчезновения,

- после окончания игры (через 30 секунд) показывается итоговый результат. Скрипт разделен на функции для генерации игрового поля, обработки кликов и завершения игрового процесса.

Скрипт `profile.js` загружает данные профиля пользователя из конфигурационного файла и формирует интерфейс страницы. Каждое поле (например, имя, группа, возраст) отображается в виде отдельного блока. Реализация показана на рисунке 5. Логика скрипта включает:

- итерацию по данным пользователя,
- создание HTML-элементов для отображения информации,
- добавление разделителей между блоками для улучшения читаемости.

Скрипт `retake.js` отвечает за функциональность формы записи на пересдачу. Логика основана на фильтрации дисциплин с неудовлетворительными оценками, указанных в конфигурационном файле. После выбора предмета пользователем данные отправляются на сервер или обрабатываются локально. Скрипт представлен на рисунке 7.

Скрипт `schedule.js` Этот скрипт реализует отображение расписания в формате календаря. Основные функции включают:

- генерацию списка дней для заданного интервала,
- сопоставление событий с датами календаря,
- создание и стилизацию элементов календаря и событий. Скрипт оптимизирован для работы с большими объемами данных, так как он использует методы фильтрации и поиска только необходимых элементов.

Скрипт `select.js` Этот скрипт управляет выпадающими списками на странице с оценками. Основные задачи:

- обновление данных о дисциплинах при выборе семестра,
- расчет среднего балла и визуальное отображение результата,
- обработка взаимодействия пользователя с интерфейсом (например, открытие/закрытие выпадающего списка).

Принципы работы клиентской логики:

- вся информация хранится в конфигурационном файле, что минимизирует необходимость дополнительных серверных запросов,
- каждый скрипт реализует конкретный функционал, что упрощает отладку и добавление новых функций.

```
const config :{} = {
  user: {
    name: 'Блантер Матвей Дмитриевич',
    avatar: 'МБ',
    login: 'blanter.m.d@edu.mirea.ru',
    gender: 'муж',
    birthdate: '27.12.2004',
    age: 19,
    group: 'ИКБ0-21-23',
    personalNumber: '45A01209',
    course: 2,
    status: 'активный',
  },
  education: {
    admissionYear: { value: 2023, label: 'Год приема' },
    formingUnit: { value: 'Институт информационных технологий', label: 'Формирующее подразделение' },
    graduatingDepartment: { value: 'Кафедра игровой индустрии', label: 'Выпускающее подразделение' },
    programType: { value: 'Бакалавриат', label: 'Вид образовательной программы' },
    specialty: { value: 'Программная инженерия (Разработка и дизайн компьютерных игр и мультимедийных приложений)', label: 'Направление подготовки (специальность)' },
    fundingType: { value: 'Сверхплановое место', label: 'Вид возмещения затрат' },
    qualification: { value: 'Бакалавр', label: 'Присваиваемая квалификация' },
    standardPeriod: { value: '4 года', label: 'Нормативный срок освоения' },
    studyForm: { value: 'очная', label: 'Форма освоения' },
    targetAdmission: { value: 'нет', label: 'Целевой прием' },
  },
  semesters: {
    1: [
      {
        name: 'Введение в профессиональную деятельность',
        date: '23.12.2023',
        teacher: 'Исаева И.А.',
        type: 'Зачет',
        grade: 'Незачет',
      },
      {
        name: 'Основы российской государственности',
        date: '25.12.2023',
        teacher: 'Иванова Е.А.',
        type: 'Зачет',
        grade: 'Зачет',
      },
    ],
  },
};
```

Рисунок 4 – Конфигурационный файл

```
document.addEventListener( type: 'DOMContentLoaded', listener: () :void => {
  const populateEducationBlock = (data, container) :void => {
    for (const key in data) {
      const { value, label } = data[key];

      const infoItem :HTMLDivElement = document.createElement( tagName: 'div');
      infoItem.classList.add('education__item');

      const infoLabel :HTMLDivElement = document.createElement( tagName: 'div');
      infoLabel.classList.add('education__label');
      infoLabel.textContent = label;

      const infoValue :HTMLDivElement = document.createElement( tagName: 'div');
      infoValue.classList.add('education__value');
      infoValue.textContent = value;

      infoItem.appendChild(infoLabel);
      infoItem.appendChild(infoValue);

      container.appendChild(infoItem);

      const divider :HTMLDivElement = document.createElement( tagName: 'div');
      divider.classList.add('education__divider');
      container.appendChild(divider);
    }

    if (container.lastChild) {
      container.lastChild.remove();
    }
  };

  populateEducationBlock(config.education, document.querySelector( selectors: '.education__body'));
});
```

Рисунок 5 – Загрузка данных в профиль

```

document.addEventListener( type: 'DOMContentLoaded', listener: () :void => {  webanet
function getNextEvent(subjects) { Show usages  webanet
    return subjects
        .map(subject => {
            return { ...subject, datetime: new Date( value: `${subject.date}T${subject.time}` ) };
        })
        .filter(subject => subject.datetime > new Date())
        .sort((a, b) => a.datetime - b.datetime)[0];
}

const nextEvent = getNextEvent(config.session.subjects);

if (nextEvent) {
    document.querySelector( selectors: '.timer__subject').textContent = `${nextEvent.controlType.toLowerCase()} no предметы "${nextEvent.name}"`;

    function updateCountdown() :void { Show usages  webanet
        const diff :number = nextEvent.datetime - new Date();

        if (diff <= 0) {
            document.querySelector( selectors: '.timer').textContent = 'Событие уже началось!';
            return;
        }

        document.querySelector( selectors: '.timer__days').textContent = Math.floor( x: diff / (1000 * 60 * 60 * 24));
        document.querySelector( selectors: '.timer__hours').textContent = Math.floor( x: (diff / (1000 * 60 * 60)) % 24);
        document.querySelector( selectors: '.timer__minutes').textContent = Math.floor( x: (diff / (1000 * 60)) % 60);
        document.querySelector( selectors: '.timer__seconds').textContent = Math.floor( x: (diff / 1000) % 60);
    }

    updateCountdown();
    setInterval(updateCountdown, timeout: 1000);
} else {
    document.querySelector( selectors: '.timer').textContent = 'Нет предстоящих событий!';
}
}
});

```

Рисунок 6 – Обратный отсчет до ближайшего зачета

```

document.addEventListener( type: 'DOMContentLoaded', listener: () :void => {  webanet
    const retakeSubjects :any[] = [];

    Object.values(config.semesters).forEach((semester) :void => {
        semester.forEach((subject) :void => {
            if (subject.grade === 'Незачет' || subject.grade === 2) {
                retakeSubjects.push(subject.name);
            }
        });
    });

    retakeSubjects.forEach((subject) :void => {
        const option :HTMLOptionElement = document.createElement( tagName: 'option');
        option.value = subject;
        option.textContent = subject;
        document.querySelector( selectors: '#subject-select').appendChild(option);
    });
});

```

Рисунок 7 – Валидация формы записи на пересдачу

3 ОПТИМИЗАЦИЯ КЛИЕНТСКОЙ ЧАСТИ ИНТЕРНЕТ-РЕСУРСА

3.1 Стилизация приложения

Для стилизации интерфейса приложения используются следующие CSS-файлы:

- global.css – содержит базовые стили для сброса стандартных стилей браузера (рисунок 8) и определения основных параметров оформления (шрифты, цвета, отступы),
- index.css – стили для главной страницы (таймер, игровое поле),
- schedule.css – оформление календаря и событий на странице расписания,
- subjects.css – стили для отображения оценок, включая цветовую маркировку,
- profile.css – оформление страницы профиля пользователя,
- retake.css – стили для формы записи на пересдачу.

Все файлы построены на основе общей концепции дизайна, что обеспечивает единообразие оформления.

```

* {
    box-sizing: border-box
}

ul, ol {
    list-style: none;
    margin: 0;
    padding: 0
}

form, p, h1, h2, h3, h4, h5, h6 {
    margin: 0;
    padding: 0;
    font-weight: normal
}

button {
    border: none
}

:focus, :active, a:focus, a:active, input::-ms-clear {
    outline: none
}

:root {
    --cs1: #000;
    --cs2: #fff;
    --csw1: #343e56;
}

body {
    font-family: Lato, sans-serif;
    margin: 0;
    height: 100vh;
    overflow-x: hidden;
    scrollbar-color: rgb(176 179 195 / 60%) transparent;
    scrollbar-gutter: stable;
    background-color: #f4f4f8;
    color: var(--csw1)
}

```

Рисунок 8 – Обнуляющие стили

3.2 Оптимизация клиентской части интернет-ресурса для стационарных и мобильных устройств

Для адаптации сайта под мобильные устройства в CSS-файлах были добавлены медиа-запросы. Пример показан на рисунках 9 и 10.

```
@media (max-width: 768px) {  
    .header__top {  
        flex-direction: column;  
        align-items: center;  
        text-align: center;  
        gap: 10px  
    }  
  
    .header__menu {  
        flex-direction: column;  
        gap: 10px  
    }  
  
    .menu {  
        flex-direction: column;  
        align-items: center;  
        gap: 10px  
    }  
  
    .menu__list {  
        flex-direction: column;  
        align-items: center;  
        gap: 10px  
    }  
  
    .menu__item {  
        width: 100%;  
        text-align: center  
    }  
  
    .menu__link {  
        width: 100%;  
        text-align: center;  
        padding: 10px;  
        font-size: 16px;  
        border-radius: 8px  
    }  
}
```

Рисунок 9 – Адаптация шапки сайта с помощью медиа-запросов

```

@media (max-width: 768px) {
  .calendar__header {
    text-align: center;
    margin-bottom: 20px;
  }

  .calendar__title {
    font-size: 16px;
  }

  .calendar__body {
    grid-template-columns: repeat(2, 1fr);
    gap: 15px;
  }

  .calendar__day {
    min-height: 80px;
    padding: 8px;
  }

  .calendar__day-date {
    font-size: 12px;
    margin-bottom: 5px;
  }

  .calendar__event {
    font-size: 10px;
    padding: 2px 4px;
  }

  .calendar__legend-list {
    flex-direction: column;
    align-items: flex-start;
    gap: 10px;
  }

  .calendar__legend-title {
    font-size: 14px;
  }

  .calendar__legend-item {
    font-size: 12px;
  }
}

```

Рисунок 10 – Адаптация календаря с помощью медиа-запросов

Благодаря адаптивности сайт корректно отображается как на больших экранах (например, на ноутбуках или персональных компьютерах), так и на мобильных устройствах. Примеры представлены на рисунках 11 и 12.

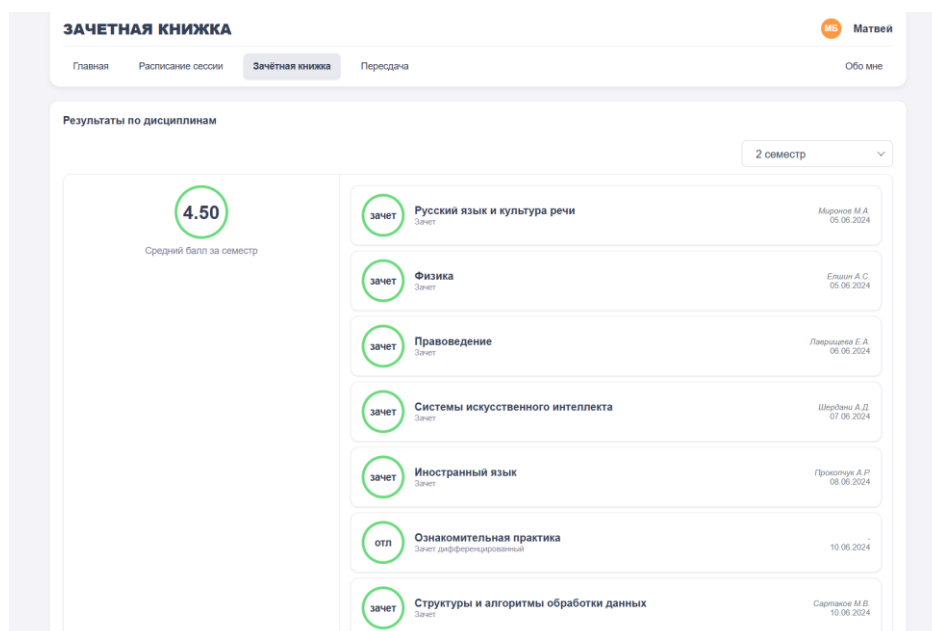


Рисунок 11 – Интерфейс сайта на ПК

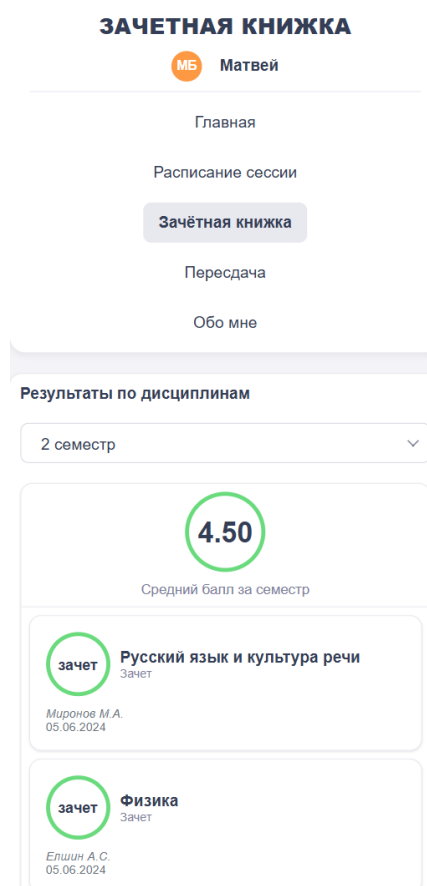


Рисунок 12 – Интерфейс сайта на телефоне

ЗАКЛЮЧЕНИЕ

В рамках курсовой работы по дисциплине "Проектирование и разработка клиентских частей интернет-ресурсов" разработана электронная зачетная книжка, представляющая собой функциональный и удобный интерфейс для пользователя. Клиентская часть имеет адаптацию под мобильные устройства, а также скрипты для интерактивности и визуальной привлекательности сайта. Таким образом, курсовая работа достигла своих целей и задач, подтверждая актуальность и важность разработки электронных решений в образовательной сфере.

Клиентская часть сайта доступна по ссылке: <http://h97768cr.beget.tech/>.

Исходный код проекта размещён в репозитории на GitHub:
https://github.com/webanetStorm/PiRKCHIR_webanet/.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Руководство по HTML/CSS/JavaScript – Статья на Хабр [Электронный ресурс]. – URL <https://habr.com/ru/articles/275729/> (дата обращения 08.10.2023).
2. JavaScript and HTML DOM Reference [Электронный ресурс]. – URL <https://www.w3schools.com/jsref/default.asp> (дата обращения 27.10.2023).