Государственное бюджетное общеобразовательное учреждение

Республиканский физико-математический лицей интернат

Партнерский итоговый проект

«Создание веб-сайта для продажи ручного труда»

Работа ученика 10 В класса

Еланского Максима Дмитриевича

ученицы 9 Г класса

Уртаевой Елизаветы Валерьевны

Руководители проекта

Молчанова Ирина

Александровна,

Макаренко Мария

Дмитриева

**ОГЛАВЛЕНИЕ**

Введение………………………………………………………………………...…3

Глава 1. Техническое задание проекта

1. Точные задачи и цели проекта………………………………………..4

Глава 2. Веб – разработка важная составляющая современного мира

1. Понятие веб – разработки……………………………………………6
2. Применение веб – разработки в проекте ……………………………6

Глава 3. Структура сайта

1. Основные возможности приложения………………..……….…….7
2. Использование HTML, CSS, Bootstrap в

проекте………………………………………...…9

Глава 4. Практическая часть. Разработка веб – сайта на Flask

1. **Разработка backend** ……………………………...11
2. **Разработка frontend** …………………………………………………...11
3. **Написание кода** …………………………………………………...11

Заключение………………………………………………………………………13

Список литературы……………………………………………..……………….13

**Введение**

В современных условиях цифровизации образования автоматизация процессов учёта учащихся становится не просто удобным инструментом, а необходимостью. Центры дополнительного образования ежедневно сталкиваются с большими объёмами данных: информация о зачислении, посещаемости, успеваемости и направлениях обучения требует систематизации и оперативного доступа.

Ручной сбор и обработка данных занимают значительное время, повышают риск ошибок и усложняют формирование отчётов. Существующие решения часто оказываются либо избыточными, либо недостаточно гибкими для специфики работы ЦДО. Данный проект призван решить эти проблемы, предложив специализированное веб-приложение для автоматизации учёта учащихся.

**Цель проекта**

Разработка веб-приложения, которое:

* Упростит внесение и хранение данных об учащихся.
* Автоматизирует формирование отчётов по ключевым показателям.
* Обеспечит удобный интерфейс для сотрудников и администраторов.
* Позволит импортировать и экспортировать данные в Excel для интеграции с внешними системами.

**Задачи**

1. **Разработка backend-части** на Flask:
   * Создание базы данных (SQLite) для хранения информации.
   * Реализация CRUD-операций (добавление, чтение, обновление, удаление данных).
   * Настройка валидации вводимых данных.
2. **Проектирование frontend-интерфейса** с использованием Bootstrap:
   * Адаптивные формы для внесения данных.
   * Инструменты поиска и фильтрации.
   * Генерация отчётов в табличном виде.
3. **Интеграция с Excel**:
   * Загрузка данных из таблиц.
   * Выгрузка отчётов для дальнейшей обработки.

**Техническое задание проекта**

##### **Назначение разработки**

* Упрощение процесса внесения и хранения данных об учащихся.
* Автоматизация формирования отчётов по различным критериям (возраст, школа, направление обучения, размещение информации об успеваемости учащихся, контроль над посещением занятий).
* Взаимодействие с Excel для импорта/экспорта данных.
* Создание удобного веб-интерфейса с использованием Bootstrap.
* Валидация (корректные данные, а также защита) данных при внесении информации об учащихся.

##### **Целевая аудитория**

* Сотрудники ЦДО it-cube «DRS», ответственные за учёт учащихся.
* Администраторы.

##### **Требования к программе**

1. **Внесение информации об учащихся:**
   * Возможность вносить данные об учащихся: ФИО, возраст, паспортные данные, школа, направление обучения, сертификат ПФДО.
   * Данные должны храниться в базе данных.
   * Валидация данных при вводе (проверка корректности возраста, уникальности сертификата ПФДО и т.д.).
2. **Запись на направление обучения:**
   * При внесении данных учащегося должна быть возможность выбрать направление обучения.
   * Данные о направлении должны быть привязаны к учащемуся.
3. **Поиск учащихся:**
   * Поиск должен осуществляться по:
     + Сертификату ПФДО.
     + ФИО (если сертификат отсутствует).
4. **Формирование отчётов:**
   * Программа должна формировать отчёты по следующим критериям:
     + Количество учащихся определённого возраста.
     + Количество учащихся из конкретной школы.
     + Количество учащихся по направлениям обучения.
5. **Импорт/экспорт данных:**
   * Возможность импорта данных из Excel в систему.
   * Возможность экспорта данных в Excel для дальнейшей обработки.
6. **Автоматическое обновление возраста:**
   * При каждом запуске программы возраст учащихся должен автоматически обновляться на основе текущей даты (с использованием библиотеки datetime).
7. **Веб-интерфейс:**
   * Приложение должно быть реализовано как веб-сайт с использованием Flask и Bootstrap.
   * Использование HTML-форм и Flask-WTF для обработки данных.
8. **Работа с JSON-файлами:**
   * Данные об учащихся должны храниться в Excel-файлах/базе данных.
   * Реализовать функции для чтения и записи в базу данных.
9. **Протокол HTTP:**
   * Приложение должно корректно обрабатывать HTTP-запросы (GET, POST, PUT, DELETE, PATCH).
10. **Валидация данных:**
    * Проверка корректности вводимых данных (например, возраст должен быть числом, сертификат ПФДО должен быть уникальным).

**Веб – разработка: Применение ее в современном мире**

#### ****1. Общее понимание веб-разработки****

Веб-разработка — это процесс создания веб-приложений и сайтов, состоящий из двух основных направлений:

* **Frontend** (клиентская часть) — то, что видят пользователи:
  + Интерфейсы (формы, кнопки, таблицы)
  + Интерактивные элементы (анимации, динамическое обновление данных)
  + Адаптивный дизайн (корректное отображение на любых устройствах)
* **Backend** (серверная часть) — "мозг" системы:
  + Обработка данных
  + Работа с базой данных
  + Бизнес-логика приложения
  + API для взаимодействия с frontend

Для нашего проекта используется стек технологий:

* **Frontend**: HTML5, CSS3, Bootstrap (для адаптивного дизайна)
* **Backend**: Python + Flask (легкий и гибкий фреймворк)
* **База данных**: SQLite (простое файловое хранение)

#### ****2. Как веб-разработка реализуется в данном проекте****

**Создание пользовательского интерфейса**

* **Формы ввода данных** об учащихся (ФИО, возраст, направление обучения)
* **Таблицы отображения** информации (списки учащихся, результаты поиска)
* **Интерактивные элементы** (фильтры для отчётов, кнопки экспорта)

**Автоматизация рутинных операций**

* Автоматическое вычисление возраста учащихся
* Мгновенное формирование отчётов
* Исключение ошибок при ручном вводе данных

**Доступность из любого места**

* Не требует установки специального ПО
* Доступ через браузер с компьютера или телефона
* Несколько сотрудников могут работать одновременно

**Гибкость системы**

* Легко добавлять новые поля (например, контакты родителей)
* Просто расширять функционал (добавить модуль посещаемости)
* Возможность интеграции с другими системами

**Структура сайта**

### **Основные возможности приложения**

#### ****1. Управление данными учащихся****

* **Добавление новых учащихся**  
  Позволяет вносить полную информацию о каждом ученике, включая ФИО, возраст, контактные данные, школу, выбранное направление обучения и номер сертификата ПФДО. Форма включает валидацию данных для предотвращения ошибок ввода.
* **Редактирование и удаление записей**  
  Дает возможность корректировать информацию об учащихся или удалять их из системы при необходимости.
* **Просмотр списка учащихся**  
  Отображает таблицу со всеми зарегистрированными учениками, поддерживает сортировку и быстрый поиск по ключевым полям (ФИО, сертификат ПФДО).

#### ****2. Формирование отчетов****

Приложение генерирует отчеты по нескольким параметрам, что упрощает анализ данных и подготовку статистики:

* **Отчет по направлениям обучения**  
  Показывает распределение учащихся по выбранным образовательным программам (например, программирование, дизайн, робототехника).
* **Отчет по школам**  
  Анализирует, из каких учебных заведений приходят ученики, помогая выявлять партнерские связи.
* **Отчет по возрасту**  
  Группирует учащихся по возрастным категориям для адаптации учебных программ.
* **Отчет по полу**  
  Предоставляет статистику по гендерному распределению учащихся.

Каждый отчет можно фильтровать (например, выбрать конкретную школу или направление) и экспортировать в Excel для дальнейшей работы.

#### ****3. Поиск и фильтрация****

* **Поиск по ФИО или сертификату ПФДО**  
  Позволяет быстро находить нужного ученика без просмотра всего списка.
* **Расширенные фильтры**  
  Дают возможность отбирать учащихся по комбинации критериев: школа + направление, возраст + пол и т.д.

#### ****4. Импорт и экспорт данных****

* **Загрузка данных из Excel**  
  Позволяет массово добавлять учащихся из готовых таблиц, экономя время при начале работы с системой или обновлении информации.
* **Выгрузка данных в Excel**  
  Поддерживает экспорт как полной базы учащихся, так и результатов отчетов для внешнего анализа или печати.

**Для чего используется HTML, CSS, Bootstrap в проекте**

#### ****1. HTML (HyperText Markup Language)****

**Что это:**  
HTML — это стандартный язык разметки для создания структуры веб-страниц. Он определяет элементы интерфейса (заголовки, формы, таблицы) и их иерархию.

**Как используется в проекте:**

* **Создание форм** для ввода данных учащихся (ФИО, возраст, направление обучения).

<form>

<input type="text" placeholder="ФИО" required>

<input type="number" placeholder="Возраст">

<select>

<option>Программирование</option>

<option>Дизайн</option>

</select>

</form>

* **Построение таблиц** для отображения списка учащихся или отчетов.
* **Организация разделов** (главная страница, отчеты, поиск).

#### ****2. CSS (Cascading Style Sheets)****

**Что это:**  
CSS — язык стилей, отвечающий за внешний вид HTML-элементов (цвета, шрифты, отступы).

**Как используется в проекте:**

* **Оформление интерфейса**:
  + Стилизация форм и кнопок.
  + Адаптация таблиц для удобного чтения.

table {

width: 100%;

border-collapse: collapse;

}

th {

background-color: #f2f2f2;

}

* **Адаптивность**: Медиа-запросы для корректного отображения на мобильных устройствах.
* **Визуальные эффекты**: Подсветка полей при ошибках валидации.

#### ****3. Bootstrap****

**Что это:**  
Bootstrap — популярный CSS-фреймворк с готовыми компонентами (кнопки, сетки, модальные окна) и утилитами для быстрой верстки.

**Как используется в проекте:**

* **Готовые компоненты**:
  + Навигационное меню.
  + Карточки для отчетов.
  + Адаптивные таблицы.

<div class="card">

<div class="card-header">Отчет по направлениям</div>

<div class="card-body">

<select class="form-control">...</select>

</div>

</div>

* **Сетка (Grid System)**:  
  Разметка страницы на колонки (например, список учащихся слева, фильтры справа).
* **Упрощение адаптивности**:  
  Автоматическое перестроение интерфейса для мобильных устройств.

**Практическая часть. Разработка веб – сайта на Flask**

#### ****Разработка backend (Flask)****

* Создание маршрутов (роутов):
  + Главная страница (/)
  + Добавление учащихся (/add\_student)
  + Просмотр списка (/students)
  + Поиск (/search)
  + Отчеты (/reports)
  + Экспорт (/export)
* Подключение базы данных (SQLite):
  + Создание таблицы students
  + Реализация CRUD-операций (добавление, чтение, обновление, удаление)
* Валидация данных (проверка возраста, уникальность сертификата)

#### ****Разработка frontend (HTML, CSS, Bootstrap)****

* Создание базового шаблона (base.html) с навигационным меню
* Верстка форм:
  + Форма добавления учащегося (add\_student.html)
  + Форма поиска (search.html)
* Отображение данных:
  + Таблица списка учащихся (students.html)
  + Страница отчетов (reports.html)
* Адаптивный дизайн (Bootstrap)

**Написание кода**

Для реализации проекта я использую Python, среда программирования – Python. Сначала сделаем функцию добавления учащегся

@app.route('/post', methods=['POST', 'GET'])  
def post():  
 form = RegistrationForm()  
 if form.validate\_on\_submit():  
 session = db\_session.create\_session()  
 if session.query(Student).filter(Student.name\_student == form.name\_student.data).first():  
 return render\_template("registration.html", form=form, message="Такой ученик уже существует")  
 user = Student(name\_student=form.name\_student.data, name\_parent=form.name\_parent.data,  
 birthday=form.birthday.data, PFDO=form.PFDO.data,  
 pasport\_number=form.pasport\_number.data, pasport\_series=form.pasport\_series.data,  
 school=form.school.data, parent\_phone=form.parent\_phone.data,  
 student\_phone=form.student\_phone.data, school\_class=form.school\_class.data,  
 adres\_of\_living=form.school.data)  
 session.add(user)  
 session.commit()  
 return redirect("/post")  
 return render\_template("registration.html", form=form)

Функция собирает данный об учащемся в форме form и записывает из в базу данных. Далее на HTML реализуем вывод списка учащихся:

<!doctype html>  
<html lang="ru">  
<head>  
 <link rel="stylesheet" href="static/css/style2.css">  
</head>  
<body>  
<table class="table">  
 <thead>  
 <tr>  
 <th>ФИО ребёнка</th>  
 <th>Дата рождения</th>  
 <th>Серия паспорта</th>  
 <th>Номер паспорта</th>  
 <th>Школа</th>  
 <th>Номер сертификата ПФДО</th>  
 <th>Телефон ученика</th>  
 <th>Класс</th>  
 <th>Адрес</th>  
 </tr>  
 </thead>  
 <tbody>  
 {% for student in students %}  
 <tr>  
 <td>{{student.name\_student}}</td>  
 <td>{{student.birthday}}</td>  
 <td>{{student.pasport\_series}}</td>  
 <td>{{student.pasport\_number}}</td>  
 <td>{{student.school}}</td>  
 <td>{{student.PFDO}}</td>  
 <td>{{student.student\_phone}}</td>  
 <td>{{student.school\_class}}</td>  
 <td>{{student.adres\_of\_living}}</td>  
 </tr>  
 {% endfor %}  
 </tbody>  
</table>  
</body>

В коде выполняется проход циклом по всем добавленным студентам и информация, добавленная о них выводится на экран.

Я показал лишь основные из реализованных функция, я выбрал их, ведь это самые част о используемые вещи в данной проекте.

**Заключение**

**В заключении хочется сказать, что реализованная стадия проекта уже работает, но стоит доделать такие делали как более удобный и понятный пользователю дизайн, также добавить некоторые функции, как формирование отчётов и редактирование данных об учащемся. Для успешного функционирования сайта его нужно перевести на сервер и постоянно обновлять в случае возникновения ошибок. Входе проведённой работы мною были изучены такие навыки разработки приложений, как: разработка Flask-приложений, улучшились знания и навыки в разработке HTML, CSS, Bootstrap. Проделанная работа мне очень понравилась и я бы хотел в будущем доработать данную систему регистрации, для ее более масштабного испорльзоваания.**

**Список литературы**

1. Официальная документация Flask: <https://flask.palletsprojects.com/>.
2. Документация Flask-WTF: <https://flask-wtf.readthedocs.io/>.
3. Руководство по работе с JSON в Python: <https://docs.python.org/3/library/json.html>.
4. Bootstrap документация: <https://getbootstrap.com/docs/>.
5. Руководство по SQLite: <https://www.sqlite.org/docs.html>.
6. Официальная документация Python: <https://docs.python.org/>.
7. Яндекс учебник <https://lms.yandex.ru/courses/1180/groups/23704/lessons/7007/materials/20630>