**РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ УПРАВЛЕНИЯ СПОРТИВНЫМ ИНВЕНТАРЁМ**

10 класс, ГБОУ Школа №1532

Участники: Жильцов Иван Дмитриевич 10A;

Жучков Иван Алексеевич 10А;

Волков Павел Сергеевич 10A;

Кангелидис Александр Игоревич 10А;

Василевский Сергей Станиславович 10A.

Руководитель: Гладникова Ксения Владимировна.

**Обоснование выбора языка программирования**

Python был выбран в качестве основного языка программирования для разработки серверной части приложения в первую очередь из-за своей простоты и читаемости кода, а ещё из-за богатой экосистемы библиотек. Также благодаря python приложение может работать на различных операционных системах без необходимости вносить изменения в код, что делает его универсальным решением для управления школьным спортивным инвентарем.

HTML был выбран для создания пользовательского интерфейса в виду доступности и простоты обучения. Также HTML хорошо совместим с другими технологиями по типу CSS.

**Обоснование выбора программных средств**

Flask был выбран в качестве микрофреймворка для разработки веб-приложения по следующим причинам: легкость и гибкость (Flask является легким и минималистичным фреймворком, который предоставляет основные функции для создания веб-приложений), быстрая разработка прототипов (Благодаря своей простоте и легкости, Flask идеально подходит для быстрой разработки MVP (минимально жизнеспособного продукта), что позволяет быстрее получать обратную связь от пользователей и корректировать функциональность приложения).

**Схема базы данных**

В нашем проекте существует несколько таблиц в базе данных. Первая таблица называется пользователи:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Тип пользователя** | **Логин** | **Пароль** | **Занятость должность/класс** | **ФИО** | **Инвентарь, который закреплён** |
| 38 | Администратор | Augustin | \*\*\*\*\*\* | Учитель физкультуры | Иванов Иван Иванович | - |
| 52 | Пользователь | sets | \*\*\*\*\*\*\* | 10 | Иванов Иван Иванович | 1 7 13 21 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID инвентаря** | **Название** | **Состояние** | **ФИО пользователя** | **Описание проблемы сломанного** |
| 87 | Козёл | Сломанный | - | Отвалилась одна из ножек козла |
| 91 | Мяч | Используемый | Иванов  Иван Иванович | - |
| 42 | Скакалка | Доступный | - | - |

Следующая таблица называется инвентарь:

Далее таблица уведомлений:

|  |  |  |
| --- | --- | --- |
| **Основной текст** | **Получатель** | **Отправитель** |
| За пользователем Максим был закреплён инвентарь, ID: 13, 14 | Все администраторы | Имя администратора |
| За вами был закреплён инвентарь, ID: 6, 7, 8 | Имя пользователя | Имя администратора |

Таблица с закупками:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Название товара** | **Категория товара** | **Цена** | **Количество** | **Поставщик** |
| 5687 | Скакалка | Спорт инвентарь | 500 | 1 | ООО ‘Спортивен’ |
| 7856 | Коврики | Для йоги | 200 | 1 | ООО ‘Спортивен’ |

И последняя таблица – для отчётов, хранит в себе единственное значение – индекс последнего отчёта:

|  |
| --- |
| Индекс последнего отчёта |
| 8 |

**Описание особенностей и аргументация выбранного типа СУБД (SQLite)**

**Встраиваемая СУБД**: SQLite не требует отдельного серверного процесса, данные хранятся в одном файле, что упрощает развертывание и управление.  
**Легковесность**: SQLite занимает мало места и потребляет минимум ресурсов, что идеально для небольших приложений.  
**Простота использования**: Не требует сложной настройки, подходит для быстрой разработки.  
**Поддержка SQL**: Полноценная поддержка стандартных SQL-запросов.  
**Кроссплатформенность**: Работает на всех популярных операционных системах.  
  
**Аргументация выбора SQLite**

**Подходит для небольших проектов**: SQLite идеально подходит для управления спортивным инвентарем, так как не требует больших объемов данных и сложной инфраструктуры.  
**Минимальные затраты на поддержку**: Отсутствие необходимости в отдельном сервере снижает затраты на администрирование.  
**Быстрое прототипирование**: SQLite позволяет быстро начать разработку без сложной настройки.  
**Локальное хранение**: Все данные хранятся в одном файле, что упрощает резервное копирование и перенос.

**Структурная и функциональная схема программного продукта:**

**Файловая структура:**

├─ **app.py**

├─ **database.db**

├─ **reports\_files**

│ ├─ *report1.txt*

├─ **static**

│ └─ *style.css*

└─ **templates**

├─ *add\_inventory.html*

*├─ admin.html*

*├─ base\_user.html*

*├─ delete\_from\_broken\_inventory.html*

*├─ delete\_inventory.html*

*├─ fix\_inventory.html*

*├─ index.html*

*├─ inventory\_all.html*

*├─ inventory\_broken.html*

*├─ inventory\_free.html*

*├─ inventory\_usefull.html*

*├─ just\_user.html*

*├─ message\_about\_break.html*

*├─ my\_inventory.html*

*├─ names\_free\_inventory.html*

*├─ notification.html*

*├─ planzakupki.html*

*├─ refix\_inventory.html*

*├─ register.html*

*├─ repeat\_delete\_inventory.html*

*├─ repeat\_fix\_inventory.html*

*├─ replace\_to\_broken\_inventory.html*

*├─ reports.html*

*├─ show\_users.html*

*└─ zayavka\_from\_user.html*

**Описание файлов:**

* app.py # Запускает проект, инициализирует базу данных, содержит все необходимые функции
* reports\_files # Папка с отчётами (в виде текстового файла)
* report1.txt # Пример отчёта
* database.db # Файл с базой данных
* static # Папка со стилями для html файлов
* style.css
* templates # Папка с html шаблонами
* index.html # Вход
* register.html # Регистрация
* base\_user # Шаблон для всех файлов ниже (кроме помеченных звёздочкой)
* admin.html # Главное окно администратора
* just\_user.html # Главное окно пользователя
* notification.html # Центр уведомлений
* # Действия администратора:
* # Просмотр инвентаря:
* inventory\_all.html # Весь
* inventory\_free.html # Свободный
* inventory\_usefull.html # Доступный
* inventory\_broken.html # Сломанный
* # Настройки инвентаря:
* add\_inventory.html # Добавить
* delete\_inventory.html # Удалить
* repeat\_delete\_inventory.html # Подтвердить удаление
* fix\_inventory.html # Закрепить за пользователем
* repeat\_fix\_inventory.html # Подтвердить закрепление
* refix\_inventory.html # Открепить от пользователя
* replace\_to\_broken\_inventory.html # Переместить в сломанный
* delete\_from\_broken\_inventory.html # Переместить из сломанного в доступный
* reports.html # Просмотр и создание отчётов
* show\_users.html # Просмотр всех пользователей
* \*planzakupki.html # Просмотр и добавление планируемых закупок
* # Действия пользователя:
* my\_inventory.html # Просмотр инвентаря пользователя
* names\_free\_inventory.html # Просмотр доступного для пользователя инвентаря
* zayavka\_from\_user.html # Создание заявки на получение инвентаря
* message\_about\_break.html # Сообщение о поломке инвентаря

**Ссылка на репозиторий с проектом**

<https://github.com/Pavel4252/moshpredprof-1532> (в файле README есть блок-схема основного алгоритма)