**Server Management System**

Система массового управления серверами Debian через SSH с веб-интерфейсом.

**Структура проекта**

server-management-system/

├── server.js # Главный файл сервера

├── package.json # Зависимости Node.js

├── .env # Переменные окружения

├── .env.example # Пример файла окружения

├── data/ # Директория для БД (создается автоматически)

│ └── servers.db # SQLite база данных

├── src/ # Исходный код

│ ├── database/

│ │ └── Database.js # Работа с базой данных

│ ├── ssh/

│ │ └── SSHManager.js # SSH подключения

│ ├── tasks/

│ │ └── TaskManager.js # Управление задачами

│ └── routes/

│ ├── servers.js # API для серверов

│ └── tasks.js # API для задач

└── public/ # Веб-интерфейс

└── index.html # Главная страница

**Установка и запуск**

**1. Клонирование и установка зависимостей**

bash

*# Создайте директорию проекта*

mkdir server-management-system

cd server-management-system

*# Скопируйте все файлы в структуре выше*

*# Затем установите зависимости*

npm install

**2. Настройка окружения**

bash

*# Скопируйте файл конфигурации*

cp .env.example .env

*# Отредактируйте .env по необходимости*

nano .env

**3. Создание директории для базы данных**

bash

mkdir -p data

**4. Запуск системы**

bash

*# Запуск в режиме разработки (с автоперезагрузкой)*

npm run dev

*# Или обычный запуск*

npm start

**5. Доступ к системе**

Откройте браузер и перейдите на http://localhost:3000

**Возможности системы**

**Управление серверами**

* ✅ Добавление серверов (IP, логин, пароль/ключ)
* ✅ Группировка серверов
* ✅ Тестирование подключений
* ✅ Получение системной информации
* ✅ Удаление серверов

**Выполнение задач**

* ✅ Смена паролей администратора
* ✅ Обновление системы (apt update/upgrade)
* ✅ Выполнение произвольных команд
* ✅ Проверка статуса системы
* ✅ Перезапуск сервисов
* ✅ Установка пакетов
* ✅ Создание пользователей
* ✅ Очистка логов

**Мониторинг**

* ✅ Real-time обновления через WebSocket
* ✅ История выполненных задач
* ✅ Детальные логи выполнения
* ✅ Статистика успешности задач

**Веб-интерфейс**

* ✅ Современный responsive дизайн
* ✅ Темная тема с градиентами
* ✅ Модальные окна для форм
* ✅ Уведомления о статусе операций
* ✅ Real-time обновления

**Конфигурация SSH**

**Аутентификация по ключу (рекомендуется)**

1. Сгенерируйте SSH ключ:

bash

ssh-keygen -t rsa -b 4096 -f ~/.ssh/server\_management

1. Скопируйте публичный ключ на целевые серверы:

bash

ssh-copy-id -i ~/.ssh/server\_management.pub user@server\_ip

1. При добавлении сервера в систему укажите путь к приватному ключу:

/home/your\_user/.ssh/server\_management

**Аутентификация по паролю**

Просто укажите пароль при добавлении сервера. Менее безопасно, но проще для тестирования.

**Примеры использования**

**1. Добавление сервера**

javascript

*// Через веб-интерфейс или API*

POST /api/servers

{

"name": "Web Server 1",

"ip": "192.168.1.10",

"username": "root",

"privateKeyPath": "/home/admin/.ssh/id\_rsa",

"port": 22,

"groupName": "web"

}

**2. Смена паролей**

javascript

*// Смена пароля root на выбранных серверах*

POST /api/tasks/predefined/change-password

{

"serverIds": [1, 2, 3],

"newPassword": "new\_secure\_password"

}

**3. Выполнение произвольной команды**

javascript

POST /api/tasks/execute

{

"name": "Check disk space",

"command": "df -h && free -h",

"serverIds": [1, 2, 3]

}

**Безопасность**

**Рекомендации:**

* 🔒 Используйте SSH ключи вместо паролей
* 🔒 Ограничьте доступ к веб-интерфейсу через reverse proxy
* 🔒 Используйте HTTPS в продакшене
* 🔒 Регулярно обновляйте систему
* 🔒 Ограничьте права пользователей SSH