**ИНФОРМАЦИОННАЯ СИСТЕМА**

**«ВЕБ-ПРИЛОЖЕНИЕ АВТО/РЕГ»**

**(ИС ВПАР)**

**РАБОЧАЯ ДОКУМЕНТАЦИЯ**

# Руководство по техническому обслуживанию

**Листов - 17**

Новосибирск, 2024

# Аннотация

Настоящий документ содержит рекомендации и порядок действий, обеспечивающих стабильное функционирование информационной системы ИС ВПАР, реализованной с использованием веб-фреймворка Django. Руководство предназначено для специалистов по техническому обслуживанию и системных администраторов.

Документ подготовлен в соответствии с РД 50-34.698-90 - в части структуры и содержания документов, и в соответствии с ГОСТ 34.201-89 - в части наименования и обозначения документов.

# Содержание

**АННОТАЦИЯ .................................................................................................................. 2**

1. **ВВЕДЕНИЕ ................................................................................................................... 4**
2. **ОБЩИЕ УКАЗАНИЯ ................................................................................................. 6**
   1. ПОРЯДОК ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ .......................................................... 6
   2. ТРЕБОВАНИЯ К ПЕРСОНАЛУ .................................................................................. 6
   3. СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ ............................................................... 7
3. **ТРЕБОВАНИЯ К ТЕХНИЧЕСКИМ СРЕДСТВАМ ............................................. 7**

3.1. Требования к техническим средствам**................................................................... 7**

1. **ОПИСАНИЕ ФУНКЦИЙ ......................................................................................... 11**
   1. ОБНОВЛЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ .................................................... 11
   2. ВЫПОЛНЕНИЕ РЕЗЕРВНОГО КОПИРОВАНИЯ ......................................................... 12
   3. ПРОВЕРКА РАБОТОСПОСОБНОСТИ СИСТЕМЫ ...................................................... 13

## 4.4 ВОССТАНОВЛЕНИЕ РАБОТОСПОСОБНОСТИ СИСТЕМЫ ПРИ АВАРИЙНЫХ

СИТУАЦИЯХ ................................................................................................................... 16

**5 ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ .............................................................. 18**

**Настоящий документ представляет собой руководство по техническому обслуживанию информационной системы «Веб-приложение Авто/Рег» (далее – ИС ВПАР).**

ИС ВПАР разработана на основе фреймворка Django и предназначена для обеспечения удобного и эффективного взаимодействия между пользователями и сервером веб-приложения. Основной функционал системы включает регистрацию, управление данными пользователей и транспортных средств, а также предоставление сопутствующих информационных услуг.

Цель документа – предоставить структурированные и детализированные указания по проведению мероприятий, необходимых для поддержания работоспособности системы, ее обновления, резервного копирования, мониторинга и восстановления после возможных сбоев. Данное руководство является основным документом для специалистов по техническому обслуживанию, администраторов баз данных и системных администраторов, обеспечивающих эксплуатацию системы.

Важность технического обслуживания обусловлена следующими факторами:

Обеспечение бесперебойной работы системы и доступности ее сервисов для пользователей.

Гарантия актуальности данных и своевременного обновления программного обеспечения.

Снижение вероятности сбоев благодаря регулярному резервному копированию и мониторингу.

Минимизация времени восстановления работоспособности системы в случае возникновения аварийных ситуаций.

Назначение ИС ВПАР

ИС ВПАР предоставляет функционал для:

Регистрации пользователей и транспортных средств.

Управления данными, включая добавление, изменение и удаление записей.

Обеспечения безопасности данных через современную систему авторизации и аутентификации.

Мониторинга операций и создания отчетности для анализа работы системы.

Система разработана с учетом современных стандартов качества и безопасности программного обеспечения. Она предназначена для работы в веб-среде и обеспечивает доступность через браузеры как на стационарных устройствах, так и на мобильных.

## 2 ОБЩИЕ УКАЗАНИЯ

### 2.1 ПОРЯДОК ТЕХНИЧЕСКОГО ОБСЛУЖИВАНИЯ

### Обслуживание включает:

### Обновление серверного программного обеспечения и приложений.

### Резервное копирование данных.

### Мониторинг и проверку работоспособности системы.

### Устранение аварийных ситуаций.

### 2.2 ТРЕБОВАНИЯ К ПЕРСОНАЛУ

### Для обслуживания ИС ВПАР специалист должен обладать:

### Знанием Django и PostgreSQL;

### Умением работать с Linux/Windows (в зависимости от ОС сервера);

### Навыками конфигурирования веб-серверов (например, Nginx, Gunicorn);

### Опыт работы с системами резервного копирования и управления логами.

### 2.3 СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

ИС – Информационная система

БД – База данных

ПО – Программное обеспечение

Django – веб-фреймворк для Python

SQL – язык структурированных запросов

## 3 ТРЕБОВАНИЯ К ТЕХНИЧЕСКИМ СРЕДСТВАМ

**3.1. Требования к техническим средствам**

Для стабильной и эффективной работы информационной системы «Веб-приложение Авто/Рег» (ИС ВПАР) необходимо соответствие используемого оборудования минимальным и рекомендуемым требованиям. Они включают характеристики серверного и пользовательского оборудования, а также требования к сетевой инфраструктуре.

**Серверное оборудование**

**Минимальные требования**

Эти требования подходят для систем с небольшим количеством пользователей (до 100 одновременных подключений):

Процессор: 2 ядра, тактовая частота от 2.4 ГГц (например, Intel Xeon E3-1220 или аналогичный).

Оперативная память (RAM): 8 ГБ.

Хранилище:

Объем – 50 ГБ для базового функционирования системы и хранения резервных копий.

Тип – SSD для повышения скорости работы с данными.

Сетевой интерфейс: 1 Гбит/с Ethernet.

Операционная система:

Linux (например, Ubuntu Server 22.04 или CentOS 8).

Windows Server 2019 (по необходимости).

**Рекомендуемые требования**

Для крупных организаций или систем с интенсивным использованием (от 100 до 1000 одновременных подключений):

Процессор: 4–8 ядер, тактовая частота от 3.0 ГГц (например, Intel Xeon E5 или AMD EPYC).

Оперативная память (RAM): 16–32 ГБ для обеспечения высокой производительности.

Хранилище:

Объем – 200 ГБ и выше, в зависимости от объемов данных.

Тип – NVMe SSD для минимизации задержек при доступе к данным.

Сетевой интерфейс: 10 Гбит/с Ethernet (для систем с высокими требованиями к скорости обработки сетевых запросов).

Операционная система: Современные версии Linux (Ubuntu Server LTS, Debian, RHEL) или Windows Server с актуальными обновлениями безопасности.

Дополнительные требования к серверу

Наличие системы бесперебойного питания (UPS) для защиты от сбоев в электроснабжении.

Использование RAID-массивов для обеспечения надежности хранения данных.

Резервирование серверного оборудования для обеспечения отказоустойчивости (например, использование виртуальных машин или кластеризации).

**Пользовательское оборудование**

Минимальные требования

Процессор: 2 ядра, тактовая частота от 2.0 ГГц.

Оперативная память: 4 ГБ.

Хранилище: 10 ГБ свободного пространства на жестком диске.

Операционная система:

Windows 10 или выше.

macOS 11 или выше.

Linux с актуальными обновлениями (например, Ubuntu 20.04).

Браузер:

Google Chrome (актуальная версия).

Mozilla Firefox (актуальная версия).

Microsoft Edge (на Chromium).

Рекомендуемые требования

Процессор: 4 ядра, тактовая частота от 3.0 ГГц.

Оперативная память: 8 ГБ.

Хранилище: SSD с объемом от 50 ГБ свободного пространства.

Операционная система:

Windows 11, macOS Ventura или Linux (с новыми ядрами).

Монитор: разрешение от Full HD (1920x1080) для удобного взаимодействия с системой.

**Сетевая инфраструктура**

Минимальные требования

Пропускная способность сети: 100 Мбит/с для серверного и пользовательского оборудования.

Задержки сети: не более 50 мс для обеспечения быстрого отклика системы.

Доступ в Интернет:

Скорость загрузки/выгрузки – не менее 10 Мбит/с.

Использование VPN для удаленного доступа к системе.

Рекомендуемые требования

Пропускная способность сети: от 1 Гбит/с для серверного оборудования и 300 Мбит/с для пользовательских устройств.

Задержки сети: до 10 мс.

Интернет: высокоскоростное подключение от 100 Мбит/с.

Фаерволы и системы предотвращения атак (IPS) для защиты от несанкционированного доступа.

**Виртуальная инфраструктура**

Для оптимизации использования ресурсов допускается виртуализация серверного оборудования:

Гипервизоры: VMware ESXi, Proxmox, Microsoft Hyper-V, VirtualBox.

Контейнеризация: Docker с использованием оркестраторов (Kubernetes).

Минимальная конфигурация виртуальных машин должна соответствовать требованиям серверного оборудования.

Данные требования обеспечивают стабильную и производительную работу ИС ВПАР, а также позволяют масштабировать инфраструктуру в зависимости от роста нагрузки на систему.

## 4 ОПИСАНИЕ ФУНКЦИЙ

### 4.1 ОБНОВЛЕНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

**Обновление Django-приложения:**

1. Перейти в каталог приложения:

bash

Копировать код

cd /path/to/project

1. Обновить зависимости:

bash

Копировать код

pip install -r requirements.txt

1. Применить миграции:

bash

Копировать код

python manage.py migrate

1. Перезапустить сервис (например, Gunicorn):

bash

Копировать код

systemctl restart gunicorn

#### 4.2 ОБНОВЛЕНИЕ БАЗЫ ДАННЫХ ИС УНП НА СЕРВЕРЕ БАЗ ДАННЫХ

**Полный бэкап выполняется еженедельно:**

bash

**Копировать код**

pg\_dump -U username -F c dbname > /backup/dbname\_full\_YYYYMMDD.dump

Инкрементальные бэкапы (если настроено): через rsync или pg\_basebackup.

4.2.2. Резервное копирование файлов приложения:

**Сохранять копии каталогов:**

Статических файлов (/static);

Медиа-файлов (/media).

**Мониторинг серверов:**

Использовать системные утилиты:

htop или top для мониторинга нагрузки.

Просмотр логов:

bash

tail -f /var/log/nginx/access.log

tail -f /path/to/project/logs/gunicorn.log

**Проверка через Django админ-панель:**

Авторизоваться в /admin.

Выполнить тестовые операции (например, создание и удаление записей).

**Проверить статус сервера:**

bash

Копировать код

systemctl status gunicorn nginx

Перезапустить сервисы.

**Восстановление БД:**

Остановить сервер приложений.

Восстановить из резервной копии:

bash

Копировать код

pg\_restore -U username -d dbname /backup/dbname\_full\_YYYYMMDD.dump

### Запуск сервера и виртуальной среды

Для корректной работы сервера и виртуальной среды системы ИС ВПАР необходимо выполнить следующие шаги. Приведенные инструкции подходят для среды разработки и тестирования, а также для развёртывания в рабочей среде.

## ****Запуск сервера****

1. **Установите зависимости**  
   Убедитесь, что все необходимые компоненты и библиотеки установлены:
   * Python (не ниже версии 3.9).
   * Система управления зависимостями **pip**.
   * СУБД (например, PostgreSQL, MySQL) в зависимости от требований приложения.
   * Дополнительные системные утилиты (например, Node.js, если требуется для фронтенда).
2. **Клонируйте репозиторий проекта**

bash

Копировать код

git clone <URL\_репозитория>

cd <директория\_проекта>

1. **Настройте окружение сервера**  
   Создайте файл конфигурации (например, .env) или измените параметры в settings.py для подключения к базе данных, указания порта и других параметров.
2. **Установите зависимости**  
   В директории проекта выполните:

bash

Копировать код

pip install -r requirements.txt

1. **Инициализируйте базу данных**  
   Выполните миграции базы данных:

bash

Копировать код

python manage.py migrate

1. **Запустите сервер**  
   Для локального тестирования:

bash

Копировать код

python manage.py runserver

Если вы используете другой сервер (например, Gunicorn или Nginx для боевого режима):

bash

Копировать код

gunicorn <имя\_приложения>.wsgi:application --bind 0.0.0.0:8000

1. **Проверьте доступность сервера**  
   Откройте браузер и перейдите по адресу:
   * **Локальный запуск**: http://127.0.0.1:8000
   * **Для удалённого доступа**: http://<IP-адрес>:<порт>

## ****2. Запуск виртуальной среды****

1. **Создайте виртуальную среду**  
   В директории проекта выполните:

bash

Копировать код

python -m venv venv

1. **Активируйте виртуальную среду**
   * **Windows**:

cmd

Копировать код

venv\Scripts\activate

* + **Linux/Mac**:

bash

Копировать код

source venv/bin/activate

1. **Проверьте активацию среды**  
   После активации виртуальной среды в консоли должен отображаться префикс (venv).
2. **Установите зависимости в виртуальной среде**

bash

Копировать код

pip install -r requirements.txt

1. **Запустите сервер из виртуальной среды**  
   Убедитесь, что виртуальная среда активна, и выполните команду:

bash

Копировать код

python manage.py runserver

1. **Отключение виртуальной среды**  
   После завершения работы отключите виртуальную среду:

bash

Копировать код

deactivate

## ****3. Автоматизация запуска****

### ****Скрипт для запуска сервера****

Создайте скрипт start\_server.sh (для Linux/Mac) или start\_server.bat (для Windows):

**Для Linux/Mac:**

bash

Копировать код

#!/bin/bash

source venv/bin/activate

python manage.py runserver

**Для Windows:**

cmd

Копировать код

@echo off

call venv\Scripts\activate

python manage.py runserver

Дайте скрипту права на выполнение (для Linux/Mac):

bash

Копировать код

chmod +x start\_server.sh

Запустите скрипт:

bash

Копировать код

./start\_server.sh

Следуя этим шагам, вы успешно развернёте сервер и виртуальную среду для работы над проектом.

## ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **версии док-та** | **Дата изменения** | **Автор изменений** | **Изменения** |
|  |  |  |  |