

# Инструкция по настройке и управлению системой голосования ЖК Аквамарин

Полное руководство по настройке хоста, базы данных, добавлению и удалению пользователей, а также остановке и перезапуску сервисов

Дата: 17 июля 2025 года

Подготовлено для демонстрации заказчику

## Введение

Данная инструкция описывает процесс настройки хоста, базы данных PostgreSQL, добавления пользователя, тестирования системы голосования для ЖК Аквамарин, удаления пользователя, а также полной остановки и повторного запуска всех сервисов. Система включает веб-интерфейс (voting.html), сервер Node.js (server.js) и базу данных PostgreSQL (voting\_system). Все команды выполняются на сервере с IP 10.211.55.11 под пользователем popcorn с доступом к sudo.

Цель: обеспечить возможность демонстрации заказчику функционала голосования, включая авторизацию, голосование за пункт «Избрать председателем Иван Ивановича» и сброс данных.

## Настройка хоста

### Установка зависимостей

Для работы веб-интерфейса и сервера требуется Node.js и http-server.

- Проверь, установлен ли Node.js:

```
node -v  
npm -v
```

Ожидаемый вывод: версии Node.js (например, v18.x.x) и npm (например, 9.x.x). Если не установлено:

```
sudo apt update  
sudo apt install -y nodejs npm
```

- Установи http-server для статических файлов:

```
sudo npm install -g http-server
```

Проверь:

```
http-server -v
```

## Запуск веб-сервера

Веб-интерфейс (`voting.html`) размещён в `/home/popcorn/Aquamarine`.

- Перейди в директорию:

```
cd /home/popcorn/Aquamarine
```

- Запусти `http-server` на порту 8080:

```
http-server -p 8080 &
```

Фоновый режим (`&`) позволяет продолжить работу в терминале.

- Проверь, что сервер работает:

```
ps aux | grep http-server  
sudo netstat -tuln | grep :8080
```

Ожидаемый вывод: процесс `http-server` и порт 8080 в состоянии LISTEN.

- Проверь доступность страницы:

```
curl http://10.211.55.11:8080/voting.html
```

Или открой в браузере: <http://10.211.55.11:8080/voting.html>. Должна отобразиться страница голосования.

## Настройка сервера Node.js

Сервер (`server.js`) обрабатывает авторизацию (`/api/login`) и голосование (`/api/vote`, `/api/vote-stats`).

- Убедись, что зависимости установлены:

```
cd /home/popcorn/Aquamarine  
npm install express pg bcrypt cors pm2
```

- Проверь `server.js` на наличие CORS:

```
const express = require('express');  
const cors = require('cors');  
const app = express();  
app.use(cors({ origin: 'http://10.211.55.11:8080' }));
```

Если CORS отсутствует, добавь строку `app.use(cors(...))` перед маршрутами.

- Запусти сервер с помощью PM2:

```
pm2 start server.js --name server
```

- Проверь статус:  
`pm2 status`  
Ожидаемый вывод: процесс server в статусе online.
- Проверь порт 3000:  
`sudo netstat -tuln | grep :3000`  
Ожидаемый вывод: порт 3000 в состоянии LISTEN.
- Проверь API:  
`curl http://10.211.55.11:3000/api/vote-stats/voteButton7`  
Ожидаемый вывод: {"for":0,"against":0,"abstain":0,"total":0}  
(если голосов нет).

## Настройка базы данных PostgreSQL

### Установка PostgreSQL

- Проверь, установлен ли PostgreSQL:  
`psql --version`  
Если не установлен:  
`sudo apt update`  
`sudo apt install -y postgresql postgresql-contrib`
- Запусти службу PostgreSQL:  
`sudo systemctl start postgresql`  
`sudo systemctl enable postgresql`
- Проверь статус:  
`sudo systemctl status postgresql`  
Ожидаемый вывод: active (running).

### Создание базы данных

- Подключись к PostgreSQL:  
`sudo -u postgres psql`
- Создай базу voting\_system (если не существует):  
`CREATE DATABASE voting_system;`

- Подключись к базе:  
`\c voting_system`
- Создай таблицы users и votes (если не существуют):

```
CREATE TABLE IF NOT EXISTS users (
  id SERIAL PRIMARY KEY,
  username TEXT NOT NULL UNIQUE,
  hashed_password TEXT NOT NULL
);
CREATE TABLE IF NOT EXISTS votes (
  id SERIAL PRIMARY KEY,
  user_id INTEGER REFERENCES users(id),
  vote_option TEXT NOT NULL,
  vote_id TEXT NOT NULL
);
```

- Проверь таблицы:

```
\d users
\d votes
```

- Выйди из psql:

```
\q
```

## Проверка подключения сервера к базе

В `server.js` должно быть настроено подключение:

```
const { Pool } = require('pg');
const pool = new Pool({
  user: 'postgres',
  host: '10.211.55.11',
  database: 'voting_system',
  password: 'your_password',
  port: 5432
});
```

Если подключение не работает, проверь логи:

```
pm2 logs
```

Обнови password в `server.js` и перезапусти:

```
pm2 restart server
```

## Добавление нового пользователя

### Генерация хеша пароля

Пароли хранятся в хешированном виде с использованием bcrypt.

- Создай скрипт для генерации хеша:

```
nano /home/popcorn/Aquamarine/generate_password.js
```

Вставь:

```
const bcrypt = require('bcrypt');
const saltRounds = 10;
const password = 'popcorn123';

bcrypt.hash(password, saltRounds, (err, hash) => {
  if (err) throw err;
  console.log(hash);
});
```

- Запусти скрипт:

```
node /home/popcorn/Aquamarine/generate_password.js
```

Скопируй хеш (например, \$2b\$10\$...).

### Добавление пользователя

- Подключись к базе:

```
sudo -u postgres psql -d voting_system
```

- Добавь пользователя popcorn:

```
INSERT INTO users (username, hashed_password) VALUES ('popcorn',
'ХЕШ_ИЗ_ПРЕДЫДУЩЕГО_ШАГА');
```

Замени ХЕШ\_ИЗ\_ПРЕДЫДУЩЕГО\_ШАГА на хеш.

- Проверь:

```
SELECT * FROM users WHERE username = 'popcorn';
```

- Выйди:

```
\q
```

### Тестирование авторизации

- Открой: <http://10.211.55.11:8080/voting.html>
- Нажми «Проголосовать» для пункта 7 («Иван Иванович»).

- Введи:
  - Логин: popcorn
  - Пароль: popcorn123
- После авторизации выбери «За» и убедись, что прогресс-бар обновился (например, «За: 1 голос (100%)»).

## Тестирование системы голосования

### Проверка голосования

- Проголосуй за «Иван Ивановича»:
  - Нажми «Проголосовать» → авторизуйся (popcorn, popcorn123) → выбери «За», «Против» или «Воздержался».
  - Проверь обновление прогресс-бара и текста (например, «За: 1 голос (100%)»).

- Проверь базу:

```
sudo -u postgres psql -d voting_system
```

```
SELECT * FROM votes WHERE vote_id = 'voteButton7' AND user_id =  
(SELECT id FROM users WHERE username = 'popcorn');
```

- Проверь API:

```
curl http://10.211.55.11:3000/api/vote-stats/voteButton7
```

Ожидаемый вывод: {"for":1,"against":0,"abstain":0,"total":1}  
(если проголосовал «За»).

### Проверка статичных голосований

Убедись, что для пунктов 1–6 кнопки «Проголосовать» отсутствуют, а результаты отображаются:

- Пункт 1: 56 за (100%), 0 против, 0 воздержался, всего 56, опрос закрыт.
- Пункт 2: 56 за (100%), 0 против, 0 воздержался, всего 56, опрос закрыт.
- Пункт 3: 53 за (95%), 3 против (5%), 0 воздержался, всего 56, опрос закрыт.
- Пункт 4: 55 за (98%), 1 против (2%), 0 воздержался, всего 56, опрос закрыт.

- Пункт 5: 55 за (98%), 1 против (2%), 0 воздержался, всего 56, опрос закрыт.
- Пункт 6: 54 за (95%), 3 против (5%), 0 воздержался, всего 57, опрос закрыт.

## Удаление пользователя и сброс голосов

### Удаление пользователя

- Подключись к базе:

```
sudo -u postgres psql -d voting_system
```

- Удали голоса пользователя popcorn:

```
DELETE FROM votes WHERE user_id = (SELECT id FROM users WHERE
username = 'popcorn');
```

- Удали пользователя:

```
DELETE FROM users WHERE username = 'popcorn';
```

- Проверь:

```
SELECT * FROM users WHERE username = 'popcorn';
SELECT * FROM votes WHERE vote_id = 'voteButton7';
```

Ожидаемый вывод: пустые результаты.

- Выйди:

```
\q
```

### Сброс голосов для демонстрации

- Сбрось все голоса для voteButton7:

```
DELETE FROM votes WHERE vote_id = 'voteButton7';
```

- Проверь страницу: <http://10.211.55.11:8080/voting.html>

- Пункт 7: «За: 0 голосов (0%)», «Против: 0 голосов (0%)», «Воздержался: 0 голосов (0%)», «Всего: 0 голосов | Опрос открыт».

- Проверь API:

```
curl http://10.211.55.11:3000/api/vote-stats/voteButton7
```

Ожидаемый вывод: {"for":0,"against":0,"abstain":0,"total":0}.

## Остановка всех сервисов

### Остановка веб-сервера

- Найди процесс http-server:  
`ps aux | grep http-server`  
Скопируй PID (например, 12345).
- Останови процесс:  
`kill -9 12345`
- Проверь, что порт 8080 освободился:  
`sudo netstat -tuln | grep :8080`  
Ожидаемый вывод: ничего.

### Остановка Node.js сервера

- Останови PM2:  
`pm2 stop server`  
`pm2 delete server`
- Проверь статус:  
`pm2 status`  
Ожидаемый вывод: процесс server отсутствует.
- Проверь порт 3000:  
`sudo netstat -tuln | grep :3000`  
Ожидаемый вывод: ничего.

### Остановка PostgreSQL

- Останови службу:  
`sudo systemctl stop postgresql`
- Проверь статус:  
`sudo systemctl status postgresql`  
Ожидаемый вывод: inactive.



## Повторный запуск сервисов

### Запуск PostgreSQL

- Запусти службу:  
`sudo systemctl start postgresql`
- Проверь статус:  
`sudo systemctl status postgresql`  
Ожидаемый вывод: active (running).
- Проверь подключение:  
`sudo -u postgres psql -d voting_system -c "\d users"`  
Ожидаемый вывод: структура таблицы users.

### Запуск Node.js сервера

- Перейди в директорию:  
`cd /home/popcorn/Aquamarine`
- Запусти сервер:  
`pm2 start server.js --name server`
- Проверь статус:  
`pm2 status`
- Проверь порт:  
`sudo netstat -tuln | grep :3000`
- Проверь API:  
`curl http://10.211.55.11:3000/api/vote-stats/voteButton7`

### Запуск веб-сервера

- Запусти http-server:  
`cd /home/popcorn/Aquamarine`  
`http-server -p 8080 &`
- Проверь:  
`ps aux | grep http-server`  
`sudo netstat -tuln | grep :8080`

- Проверь страницу:

```
curl http://10.211.55.11:8080/voting.html
```

Или открой: <http://10.211.55.11:8080/voting.html>.

## Демонстрация заказчику

- Открой страницу: <http://10.211.55.11:8080/voting.html>.
- Покажи статичные результаты пунктов 1–6 (без кнопок «Проголосовать»).
- Добавь пользователя popcorn (раздел 4).
- Авторизуйся (popcorn, popcorn123), проголосуй за «Иван Ивановича», покажи обновление прогресс-бара.

- Покажи базу:

```
sudo -u postgres psql -d voting_system -c "SELECT * FROM users  
WHERE username = 'popcorn';"  
sudo -u postgres psql -d voting_system -c "SELECT * FROM votes  
WHERE vote_id = 'voteButton7';"
```

- Сбрось голоса и удали пользователя (раздел 5) для повторной демонстрации.

## Решение проблем

- **Страница не грузится:** Проверь http-server:

```
ps aux | grep http-server  
sudo netstat -tuln | grep :8080
```

Перезапусти: `http-server -p 8080 &`.

- **Ошибка авторизации:** Проверь логи:

```
pm2 logs
```

Убедись, что `server.js` использует правильный столбец (`hashed_password`) и хеш.

- **Ошибка CORS:** Добавь в `server.js`:

```
app.use(cors({ origin: 'http://10.211.55.11:8080' }));
```

Перезапусти: `pm2 restart server`.

- **Ошибка базы данных:** Проверь подключение:

```
sudo -u postgres psql -d voting_system
```

Убедись, что `server.js` использует правильный `host`, `user`, `password`.