# 🎵 Music-U-Scheduler - Fresh Install Guide

This guide provides complete instructions for setting up the Music-U-Scheduler from scratch.

## Quick Install (Recommended)

### One-Command Install

```
curl -fsSL https://raw.githubusercontent.com/dfultonthebar/Music-U-Scheduler/main/
quick-install.sh | bash
```

### Manual Install Steps

#### 1. System Requirements

- **Operating System**: Linux (Ubuntu 20.04+ recommended), macOS, or Windows with WSL
- **Node.js**: Version 20 or higher
- **Python**: Version 3.11 or higher
- **Database**: PostgreSQL 14+
- **Memory**: Minimum 2GB RAM
- **Storage**: Minimum 5GB free space

#### 2. Clone Repository

```
git clone https://github.com/dfultonthebar/Music-U-Scheduler.git
cd Music-U-Scheduler
```

#### 3. Run Installation Script

```
chmod +x install.sh
./install.sh
```

#### 4. Start the Application

```
./start-app.sh
```

## Access Information

### URLs

- **Frontend**: http://localhost:3000
- **Backend API**: http://localhost:8001
- **API Documentation**: http://localhost:8001/docs

### Default Login Credentials

- **Admin Email**: admin@musicu.com
- **Admin Password**: MusicU2025

### Test Account (Hidden from Users)

- **Test Email**: john@doe.com
- **Test Password**: johndoe123
- **Role**: Admin

## Post-Installation Setup

### 1. Change Default Passwords

```
# Access admin panel and change default passwords
curl -X POST http://localhost:8001/auth/change-password \
  -H "Content-Type: application/json" \
  -d '{"old_password": "MusicU2025", "new_password": "YourNewPassword"}'
```

### 2. Configure Email Settings

1. Log into admin panel at http://localhost:3000/admin
2. Navigate to System Settings > Email Configuration
3. Enter your SMTP server details

### 3. Customize System Settings

1. Update organization name and logo
2. Configure backup schedules
3. Set up user roles and permissions

## System Architecture

```
Music-U-Scheduler/
├── backend/                  # FastAPI backend
│   ├── app/                  # Main application
│   ├── migrations/           # Database migrations
│   └── requirements.txt      # Python dependencies
├── frontend/                 # Next.js frontend
│   ├── app/                  # App router pages
│   ├── components/           # React components
│   └── package.json          # Node dependencies
├── install.sh                # Main installer
├── start-app.sh              # Application launcher
└── docker-compose.yml        # Docker configuration
```

## Installation Scripts

### install.sh

Main installation script that:
- Checks system requirements
- Installs dependencies
- Sets up databases
- Configures environment
- Builds applications

### start-app.sh

Application launcher that:
- Starts PostgreSQL
- Launches FastAPI backend
- Starts Next.js frontend
- Sets up reverse proxy

### update-system.sh

System updater that:
- Pulls latest changes
- Updates dependencies
- Runs database migrations
- Restarts services

## Database Setup

### PostgreSQL Configuration

```
-- Create database and user
CREATE DATABASE musicu_scheduler;
CREATE USER musicu_user WITH ENCRYPTED PASSWORD 'secure_password';
GRANT ALL PRIVILEGES ON DATABASE musicu_scheduler TO musicu_user;
```

### Environment Variables

```
# Database
DATABASE_URL=postgresql://musicu_user:secure_password@localhost/musicu_scheduler

# Security
SECRET_KEY=your-secret-key-here
JWT_SECRET=your-jwt-secret-here

# Email
SMTP_SERVER=smtp.your-provider.com
SMTP_PORT=587
SMTP_USERNAME=your-email@example.com
SMTP_PASSWORD=your-email-password
```

## Troubleshooting

### Common Issues

#### Port Already in Use

```
# Kill processes on ports 3000 and 8001
sudo fuser -k 3000/tcp
sudo fuser -k 8001/tcp
```

### Database Connection Issues

```
# Check PostgreSQL status
sudo systemctl status postgresql

# Restart PostgreSQL
sudo systemctl restart postgresql
```

### Permission Issues

```
# Fix file permissions
chmod +x *.sh
chmod -R 755 frontend/
chmod -R 755 backend/
```

### Node.js Version Issues

```
# Install Node.js 20 using nvm
curl -o- https://raw.githubusercontent.com/nvm-sh/nvm/v0.39.0/install.sh | bash
source ~/.bashrc
nvm install 20
nvm use 20
```

## Log Files

- **Backend logs**: `logs/backend.log`
- **Frontend logs**: `logs/frontend.log`
- **Database logs**: `/var/log/postgresql/`
- **Nginx logs**: `/var/log/nginx/`

# Backup and Recovery

## Create Backup

```
# Database backup
pg_dump musicu_scheduler > backup_$(date +%Y%m%d_%H%M%S).sql

# Full system backup
tar -czf musicu_backup_$(date +%Y%m%d).tar.gz Music-U-Scheduler/
```

## Restore Backup

```
# Database restore
psql musicu_scheduler < backup_file.sql

# System restore
tar -xzf musicu_backup.tar.gz
```

# Security Considerations

## Production Setup

1. **Change all default passwords**
2. **Use HTTPS with SSL certificates**
3. **Configure firewall rules**
4. **Regular security updates**
5. **Database access restrictions**
6. **Backup encryption**

## SSL Configuration

```
# Generate SSL certificate
sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 \
  -keyout /etc/ssl/private/musicu.key \
  -out /etc/ssl/certs/musicu.crt
```

# System Monitoring

## Health Checks

```
# Check application status
curl http://localhost:3000/api/health
curl http://localhost:8001/health

# Check database connection
psql -h localhost -U musicu_user -d musicu_scheduler -c "SELECT 1;"
```

## System Metrics

- CPU usage monitoring
- Memory usage tracking
- Database performance
- Network connectivity
- Disk space monitoring

# Updates and Maintenance

## Regular Updates

```
# Update system packages
sudo apt update && sudo apt upgrade

# Update application
git pull origin main
./update-system.sh
```

## Maintenance Tasks

- Weekly database optimization
- Monthly backup verification

- Quarterly security audits
- Log rotation and cleanup

# Support and Documentation

## Getting Help

1. **Documentation**: Full docs at `/docs/`
2. **API Reference**: http://localhost:8001/docs
3. **GitHub Issues**: Create issue for bugs
4. **Community**: Discord/Slack for discussions

## Contributing

1. Fork the repository
2. Create feature branch
3. Make changes and test
4. Submit pull request
5. Follow coding standards

---

**Last Updated**: August 2025
**Version**: 3.0.0
**Compatibility**: Linux, macOS, Windows WSL