



Music U Scheduler - Download & Setup Guide



Quick Start (One-Line Install)

For the fastest setup, use our automated installer:

```
# Download and run the installer in one command
curl -fsSL https://raw.githubusercontent.com/dfultonthebar/Music-U-Scheduler/main/install.sh | bash
```

OR if you prefer to download first:

```
# Download the installer
wget https://raw.githubusercontent.com/dfultonthebar/Music-U-Scheduler/main/install.sh
chmod +x install.sh
./install.sh
```



Prerequisites

Before starting, ensure your system has:

Ubuntu/Debian Systems:

```
sudo apt update
sudo apt install -y python3 python3-pip python3-venv nodejs npm git curl
```

CentOS/RHEL Systems:

```
sudo yum install -y python3 python3-pip nodejs npm git curl
# or for newer versions
sudo dnf install -y python3 python3-pip nodejs npm git curl
```

macOS:

```
# Install Homebrew if not already installed
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

# Install dependencies
brew install python3 node git
```



Manual Download & Setup

If you prefer manual installation:

Step 1: Clone the Repository

```
# Clone the repository to your desired location
git clone https://github.com/dfultonthebar/Music-U-Scheduler.git
cd Music-U-Scheduler
```

Step 2: Backend Setup (Python/FastAPI)

```
# Create Python virtual environment
python3 -m venv music-u-env

# Activate virtual environment
source music-u-env/bin/activate # Linux/Mac
# OR for Windows: music-u-env\Scripts\activate

# Upgrade pip and install dependencies
pip install --upgrade pip
pip install -r requirements.txt
```

Step 3: Frontend Setup (Next.js)

```
# Navigate to frontend directory
cd app

# Install Node.js dependencies (with conflict resolution)
npm install --legacy-peer-deps

# OR if you prefer yarn:
yarn install
```

Step 4: Environment Configuration

```
# Copy environment template
cp .env.example .env

# Edit the .env file with your settings
nano .env
```

Step 5: Database Setup

```
# Initialize the database
python init_admin.py

# Run database migrations
alembic upgrade head
```

Step 6: Start the Services

Option A: Start All Services Together

```
chmod +x start-all.sh
./start-all.sh
```

Option B: Start Services Individually

Terminal 1 (Backend):

```
./start-backend.sh
```

Terminal 2 (Frontend):

```
./start-frontend.sh
```

Access Your Application

Once running, access the application at:

- **Frontend:** <http://localhost:3000>
- **Backend API:** <http://localhost:8000>
- **API Documentation:** <http://localhost:8000/docs>

Default Login Credentials

```
Username: admin  
Password: admin123
```

 **IMPORTANT:** Change the default password immediately after first login!

Service Management

The application includes service management scripts:

```
# Start all services  
./start-all.sh  
  
# Manage individual services  
./manage-services.sh
```

Testing Your Installation

Run the comprehensive test to verify everything works:

```
# Run installation test  
./test_comprehensive.py
```

Project Structure

```

Music-U-Scheduler/
├── app/                # Next.js Frontend
│   ├── components/    # React components
│   ├── pages/         # Next.js pages
│   ├── lib/           # Utility libraries
│   └── styles/         # CSS/Tailwind styles
├── alembic/           # Database migrations
├── static/            # Static files
├── templates/         # HTML templates
├── install.sh         # Automated installer
├── requirements.txt    # Python dependencies
├── pyproject.toml     # Python project config
└── README.md          # Project documentation

```

Troubleshooting

Common Issues:

1. ESLint Conflicts

```

cd app
npm install --legacy-peer-deps

```

2. Python Virtual Environment Issues

```

rm -rf music-u-env
python3 -m venv music-u-env
source music-u-env/bin/activate
pip install --upgrade pip
pip install -r requirements.txt

```

3. Port Already in Use

```

# Kill processes on ports 3000 and 8000
sudo lsof -t -i:3000 | xargs kill -9
sudo lsof -t -i:8000 | xargs kill -9

```

4. Database Issues

```

# Reset database
rm -f app.db
python init_admin.py
alembic upgrade head

```

5. Node.js Version Issues

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

# Install and use Node.js 18+
nvm install 18
nvm use 18
```

Updating Your Installation

To update to the latest version:

```
git pull origin main
source music-u-env/bin/activate
pip install -r requirements.txt
cd app && npm install --legacy-peer-deps
```

Support

If you encounter issues:










1. Check the logs in the `.logs` directory
2. Review the [TROUBLESHOOTING.md](#) (TROUBLESHOOTING.md) file
3. Create an issue on GitHub: <https://github.com/dfultonthebar/Music-U-Scheduler/issues>

Next Steps

After installation:

1. **Change default credentials**
2. **Configure email settings** (for notifications)
3. **Set up your instructors and students**
4. **Customize the application settings**
5. **Create your first lesson schedule**

Features Available

-  User authentication and authorization
-  Admin dashboard with user management
-  Instructor dashboard with lesson scheduling
-  Student portal for booking lessons
-  Real-time notifications
-  Responsive design for mobile and desktop
-  RESTful API with documentation
-  Database migrations and backups
-  Service management and monitoring

 **Enjoy using Music U Scheduler!**

For more detailed documentation, visit our [GitHub repository](https://github.com/dfultonthebar/Music-U-Scheduler) (<https://github.com/dfultonthebar/Music-U-Scheduler>).