📦 Installation Guide

This document provides multiple installation methods for the Audience Response System, ranging from automated installers to manual setup.

🚀 Quick Installation (Recommended)

Windows Users

1. **Download** `install-windows.bat`
2. **Right-click** the file and select **"Run as administrator"**
3. **Follow the prompts** - the installer will handle everything automatically
4. **Double-click** the desktop shortcut when installation completes

Mac Users

1. **Download** `install-mac.sh`
2. **Open Terminal** and navigate to the download folder
3. **Run**: `chmod +x install-mac.sh && ./install-mac.sh`
4. **Follow the prompts** - the installer will use Homebrew to install dependencies
5. **Double-click** "Audience Response System" on your desktop

Linux Users

1. **Download** `install-linux.sh`
2. **Open Terminal** and navigate to the download folder
3. **Run**: `chmod +x install-linux.sh && ./install-linux.sh`
4. **Follow the prompts** - the installer will detect your distribution automatically
5. **Use the desktop shortcut** or run the startup script

🖥️ GUI Installer (Cross-Platform)

For users who prefer a graphical installer:

Prerequisites

- Python 3.6+ with tkinter support
- Internet connection

Installation

1. **Download** `install.py`
2. **Run**: `python3 install.py` (or double-click on Windows)
3. **Use the GUI** to configure installation options
4. **Click Install** and monitor the progress

The GUI installer provides:
- ✅ Visual progress tracking
- ✅ Installation logs
- ✅ Customizable installation directory

- ✅ Optional component selection
- ✅ Error handling and troubleshooting

⚙️ Manual Installation

For advanced users or custom setups:

Step 1: Install Dependencies

Windows:
- Node.js 18+: https://nodejs.org
- Git: https://git-scm.com
- PostgreSQL 14+: https://postgresql.org

Mac (with Homebrew):

```
brew install node git postgresql@14
brew services start postgresql@14
```

Linux (Ubuntu/Debian):

```
sudo apt update
sudo apt install nodejs npm git postgresql postgresql-contrib
sudo systemctl start postgresql
sudo systemctl enable postgresql
```

Step 2: Download Application

```
git clone https://github.com/benny2744/ARS-BZ.git
cd ARS-BZ/app
```

Step 3: Setup Environment

```
# Install dependencies
npm install -g yarn
yarn install

# Copy environment template
cp .env.example .env

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

Step 4: Database Setup

```
# Create database
createdb ars_database

# Initialize schema
npx prisma migrate deploy
npx prisma db seed
```

Step 5: Build & Run

```
# Build for production
yarn build

# Start the application
yarn start
```

🔧 Configuration Options

Environment Variables (.env)

Variable	Description	Default
`DATABASE_URL`	PostgreSQL connection string	Required
`NEXTAUTH_URL`	Application URL	`http://localhost:3000`
`NEXTAUTH_SECRET`	Authentication secret	Required
`NODE_ENV`	Environment mode	`development`
`MAX_FILE_SIZE`	Max upload size (bytes)	`10485760` (10MB)
`ALLOWED_FILE_TYPES`	Allowed file extensions	`.jpg,.png,.pdf`

Database Configuration

Local PostgreSQL:

```
DATABASE_URL="postgresql://username:password@localhost:5432/ars_database"
```

Cloud PostgreSQL (recommended for production):

```
DATABASE_URL="postgresql://user:pass@host:5432/db?sslmode=require"
```

🌐 Network Setup

Local Network Access

To allow other devices on your network to access the system:

1. **Update NEXTAUTH_URL** in `.env`:
   `env
   NEXTAUTH_URL="http://YOUR-IP-ADDRESS:3000"`

2. **Configure Firewall:**
   - **Windows**: Allow port 3000 through Windows Defender Firewall
   - **Mac**: System Preferences → Security & Privacy → Firewall
   - **Linux**: `sudo ufw allow 3000`

3. **Find your IP address:**
   - **Windows**: `ipconfig`
   - **Mac/Linux**: `ifconfig` or `ip addr show`

Production Deployment

For production use, consider:
- Using a reverse proxy (nginx, Apache)
- Setting up SSL certificates
- Using a cloud database service
- Implementing proper backup strategies

🔒 Security Considerations

Essential Security Steps

1. **Change default secrets:**
   `bash`
   `# Generate a secure secret`
   `openssl rand -base64 32`

2. **Use environment-specific URLs:**
   ```env
   # Development
   NEXTAUTH_URL="http://localhost:3000"

# Production
NEXTAUTH_URL="https://yourdomain.com"
```

1. **Secure database connections:**
   `env`
   `DATABASE_URL="postgresql://user:pass@host:5432/db?sslmode=require"`

2. **Configure file upload limits:**
   `env`
   `MAX_FILE_SIZE=10485760  # 10MB`
   `ALLOWED_FILE_TYPES=".jpg,.jpeg,.png,.gif,.pdf"`

🚨 Troubleshooting

Common Issues

Node.js version errors:

```
# Check version
node --version  # Should be 18.0.0 or higher

# Update using package manager
# Windows: Download from nodejs.org
# Mac: brew upgrade node
# Linux: Use NodeSource repository
```

Database connection errors:

```
# Test PostgreSQL connection
psql -h localhost -U postgres -d ars_database

# Check if PostgreSQL is running
# Windows: Services.msc
# Mac: brew services list | grep postgresql
# Linux: sudo systemctl status postgresql
```

Port already in use:

```
# Find process using port 3000
# Windows: netstat -ano | findstr 3000
# Mac/Linux: lsof -i :3000

# Kill process and restart
```

Permission errors (Linux/Mac):

```
# Fix file permissions
chmod -R 755 /path/to/installation

# Fix ownership
sudo chown -R $USER:$USER /path/to/installation
```

Installation Logs

Check installation logs for detailed error information:
- **Windows**: `%USERPROFILE%\AudienceResponseSystem\.logs\`
- **Mac/Linux**: `~/AudienceResponseSystem/.logs/`

Getting Help

1. **Check the FAQ** in README.md
2. **Review troubleshooting** in DEPLOYMENT.md
3. **Search existing issues** on GitHub
4. **Create a new issue** with:
   - Operating system and version
   - Installation method used
   - Error messages and logs
   - Steps to reproduce

📋 System Requirements

Minimum Requirements

- **OS**: Windows 10, macOS 10.15, or Linux (Ubuntu 18.04+)
- **RAM**: 2GB available memory
- **Storage**: 1GB free disk space
- **Network**: Internet connection for initial setup
- **Database**: PostgreSQL 12+ (can be installed automatically)

Recommended Requirements

- **OS**: Latest versions of Windows 11, macOS 12+, or Ubuntu 20.04+
- **RAM**: 4GB+ available memory
- **Storage**: 2GB+ free disk space (for logs and uploads)
- **Network**: Stable internet connection and local network access
- **Database**: PostgreSQL 14+ with SSL support

✅ Post-Installation Checklist

After installation, verify these items:

- [ ] Application starts without errors
- [ ] Database connection successful
- [ ] Admin account can be created
- [ ] Sessions can be created and joined
- [ ] File uploads work correctly
- [ ] Real-time updates function properly
- [ ] Desktop shortcuts work
- [ ] Network access configured (if needed)
- [ ] Firewall rules configured
- [ ] Backup strategy implemented

🔄 Updates

Automated Updates

The installers include update functionality:
- **Windows**: Re-run `install-windows.bat`
- **Mac/Linux**: Re-run the installation script
- **GUI**: Use the “Update” option in the installer

Manual Updates

```
cd /path/to/ARS-BZ
git pull origin main
cd app
yarn install
yarn build
```

Need additional help? Visit our GitHub repository (https://github.com/benny2744/ARS-BZ) or check the troubleshooting guide (DEPLOYMENT.md#troubleshooting).