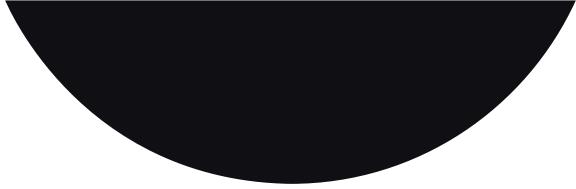
Task Manager - Deployment Guide

This application is production-ready and can be deployed to various cloud platforms. Here are the deployment options:

Quick Deploy Options

1. Railway (Recommended - Free Tier Available)





Steps:

- 1. Click the Railway button above or visit Railway.app (https://railway.app)
- 2. Connect your GitHub account and fork this repository
- 3. Deploy directly from GitHub

- 4. Railway will automatically:
- Install dependencies
- Build the frontend
- Start the backend server
- Provide a public URL

Configuration:

- No additional configuration needed
- SQLite database persists automatically
- HTTPS enabled by default

2. Render.com (Free Tier Available)

- 1. Visit Render.com (https://render.com)
- 2. Create a new Web Service
- 3. Connect your GitHub repository
- 4. Use these settings:
 - Build Command: cd client && npm ci && npm run build && cd ../server && npm ci
 - Start Command: cd server && npm start
 - **Environment:** Node
- 5. Set environment variables (see render.yaml)

3. Fly.io

- 1. Install Fly CLI: curl -L https://fly.io/install.sh | sh
- 2. Login: fly auth login
- 3. Clone and deploy:

```
bash
  git clone <your-repo>
  cd task-manager
  fly launch
  fly deploy
```

4. Docker Deployment

Using Docker Compose (Recommended):

```
git clone <your-repo>
cd task-manager
docker-compose up -d
```

Manual Docker:

```
docker build -t task-manager .
docker run -d -p 3001:3001 -v $(pwd)/data:/app/data task-manager
```

5. VPS/Server Deployment

Prerequisites:

- Node.js 18+
- PM2 (recommended for process management)

Steps:

```
git clone <your-repo>
cd task-manager
npm run install:all
npm run build

# Using PM2
npm install -g pm2
pm2 start server/index.js --name task-manager
pm2 startup
pm2 save

# Setup nginx reverse proxy (optional)
# Copy and modify nginx.conf for your domain
```

Environment Configuration

Required Environment Variables

Server (.env):

```
NODE_ENV=production
PORT=3001
JWT_SECRET=your-super-secure-jwt-secret-here
BCRYPT_ROUNDS=12
DATABASE_PATH=./data/database.sqlite
ALLOWED_ORIGINS=https://your-domain.com
```

Client (automatic based on environment):

- Development: Uses localhost:3001
- Production: Uses relative paths (/api)

Database

- · Uses SQLite for simplicity and portability
- Database file persists in /data directory
- Default admin user: admin / admin123
- Automatically seeds database on first startup

Features

- ▼ Fully Production Ready
- **✓ User Authentication** (JWT-based)
- Kanban Board with drag-and-drop
- Real-time updates via Socket.io
- Responsive design (works on mobile)
- Security headers and rate limiting
- ▼ SQLite database (zero configuration)
- Docker support
- HTTPS ready

Testing Deployment

After deployment, test these features:

- 1. Visit your deployed URL
- 2. Login with: admin / admin123
- 3. Create tasks and move them between columns
- 4. Test responsiveness on mobile
- 5. Verify real-time updates work

Troubleshooting

Common Issues:

- CORS errors: Update ALLOWED ORIGINS environment variable
- Database not persisting: Ensure /data directory is mounted properly
- Build fails: Check Node.js version (18+ required)
- Socket.io not working: Verify WebSocket support on your platform

Need help? Check the logs:

- Railway: Dashboard → Deployment → Logs
- Render: Dashboard → Service → Logs
- Fly.io: fly logs
- Docker: docker logs <container-name>

Security Notes

- ₽ IMPORTANT: Change the JWT SECRET in production!
- A Database: SQLite file contains user data ensure proper backups
- All platforms above provide HTTPS automatically
- Rate Limiting: Built-in protection against abuse