🚀 Railway.app Deployment Guide - Ultra Simple!

Step 1: Prepare Your Files

Create a folder with these files:

```
uniform-dashboard/
├── server.js
├── package.json
├── public/
│   └── index.html (your React build)
└── README.md
```

Step 2: Setup Railway MySQL Database

1. Go to railway.app and sign in with GitHub
2. Click **"New Project"**
3. Select **"Deploy MySQL"**
4. Wait for MySQL to deploy (takes 1-2 minutes)
5. Click on the MySQL service
6. Go to **"Variables"** tab
7. **COPY** these variables (you'll need them):
    - `MYSQLHOST`
    - `MYSQLPORT`
    - `MYSQLUSER`
    - `MYSQLPASSWORD`
    - `MYSQLDATABASE`

Step 3: Deploy Your Application

Method A: GitHub (Recommended)

1. Create a GitHub repository
2. Upload all your files to the repository
3. Go back to Railway dashboard

4. Click **"New" → "GitHub Repo"**

5. Select your repository

6. Railway will auto-detect it's a Node.js app!

Method B: Railway CLI (Alternative)

```bash
# Install Railway CLI
npm i -g @railway/cli

# Login to Railway
railway login

# Initialize project
railway init

# Deploy
railway up
```

Step 4: Configure Environment Variables

1. In Railway dashboard, click your deployed service

2. Go to **"Variables"** tab

3. Click **"New Variable"**

4. Add these variables one by one (use values from MySQL service):

```
MYSQLHOST = [value from MySQL service]
MYSQLPORT = [value from MySQL service]
MYSQLUSER = [value from MySQL service]
MYSQLPASSWORD = [value from MySQL service]
MYSQLDATABASE = [value from MySQL service]
PORT = 3000
```

Step 5: Connect Services

1. Click on your app service

2. Go to **"Settings"** tab

3. Under **"Service"**, find **"Networking"**

4. Copy the **public domain** (e.g., `your-app.up.railway.app`)

Step 6: Build React Frontend

Since you're using React, you need to build it first:

```bash
# In your React project folder
npm install
npm run build
```

Then copy the `build` folder contents to `public` folder in your server directory.

Step 7: Test Your Deployment

1. Visit your Railway domain: `https://your-app.up.railway.app`
2. Check health: `https://your-app.up.railway.app/health`
3. Test API: `https://your-app.up.railway.app/api/columns`

🎉 You're Live!

Your dashboard is now deployed and accessible worldwide!

Common Issues & Solutions

Issue 1: "Cannot connect to database"

Solution: Make sure all MySQL environment variables are correctly copied from the MySQL service to your app service.

Issue 2: "Port already in use"

Solution: Railway automatically assigns ports. Don't hardcode port 3000 in production. Use `process.env.PORT`.

Issue 3: "502 Bad Gateway"

Solution: Check logs in Railway dashboard. Usually means the app crashed on startup. Check your dependencies are installed.

Issue 4: "File upload not working"

Solution: Railway has file size limits. Keep uploads under 10MB and use memory storage (already configured).

Railway Dashboard Features

Viewing Logs

1. Click your service
2. Click **"Deployments"** tab
3. Click **"View Logs"**

Setting up Custom Domain

1. Go to **"Settings"**
2. Click **"Domains"**
3. Click **"Custom Domain"**
4. Follow DNS setup instructions

Monitoring

Railway automatically monitors:

- CPU usage
- Memory usage
- Request count
- Response times

Cost Information

- **MySQL Database**: ~$5/month
- **Web Service**: Starts free, then ~$5/month
- **Total**: ~$10/month for production use

Free Tier: Railway gives $5 free credit monthly!

Quick Commands Reference

```
bash
```

```
# View logs
railway logs

# Link to project
railway link

# Add MySQL plugin
railway add

# Deploy changes
railway up

# Open deployed app
railway open
```

File Structure on Railway

```
/app
├── node_modules/
├── public/
│   └── index.html
├── server.js
├── package.json
└── package-lock.json
```

Environment Variables Template

Create a `.env` file locally for testing (don't commit this!):

```
env

MYSQLHOST=localhost
MYSQLPORT=3306
MYSQLUSER=root
MYSQLPASSWORD=yourpassword
MYSQLDATABASE=uniform_db
PORT=3000
```

Auto-Deploy Setup

1. Connect GitHub repository to Railway
2. Every `git push` to main branch will auto-deploy
3. Railway shows deployment status
4. Rollback available if needed

Support & Resources

- Railway Docs: https://docs.railway.app
- Railway Discord: https://discord.gg/railway
- GitHub Issues: Create issues in your repo

Security Checklist ✅

□ MySQL password is strong
□ Environment variables are not in code
□ `.env` file is in `.gitignore`
□ API endpoints have rate limiting (add if needed)
□ File upload size is limited
□ CORS is configured properly

Next Steps After Deployment

1. Test all features thoroughly
2. Add user authentication (if needed)
3. Setup automated backups for MySQL
4. Configure custom domain
5. Setup monitoring alerts
6. Add API documentation

Need Help?

- Check Railway logs first
- Review environment variables
- Test MySQL connection separately
- Contact Railway support (very responsive!)

Your app will be live at: `https://[your-project-name].up.railway.app`

Good luck! 🚀