

# SINGLE-DEPLOYMENT DOCUMENT

## React (Vite) × 5 Frontends + Compiled Node Backend

### GOAL (WHAT WE ARE BUILDING)

You want **ONE** deployable project that:

- Has **1 backend** (Node.js) → **compiled**
- Has **5 frontends** (Vite) served at:
  - `/` → **User**
  - `/vendor` → **Vendor**
  - `/driver` → **Driver**
  - `/zone` → **Zone**
  - `/admin` → **Admin**
- Uses **clean URLs** (no `#`)
- Needs **NO NGINX** config
- Works on **aaPanel** / any server
- Can be run by user using:

```
node server.js
```

or

```
npm run start
```

### BEST WORDING FOR ROLES (RECOMMENDED)

Route	Role Name (Best Practice)
<code>/</code>	User / Customer
<code>/vendor</code>	Vendor (correct & standard)
<code>/driver</code>	Driver
<code>/zone</code>	Operations / Zone Manager
<code>/admin</code>	Administrator

✔ These terms are **industry-standard**, scalable, and clear.



## FINAL PROJECT STRUCTURE

```
project-root/
├─ server.js           # compiled backend (final output)
├─ package.json
├─ .env
├─ frontend/
│   ├─ user/
│   │   └─ dist/
│   ├─ vendor/
│   │   └─ dist/
│   ├─ driver/
│   │   └─ dist/
│   ├─ zone/
│   │   └─ dist/
│   └─ admin/
│       └─ dist/
└─ public/
    └─ uploads/
```

👉 Only these files are deployed

### ◆ STEP 1: CONFIGURE EACH VITE FRONTEND

#### 1 Set correct `base` in each Vite project

User app ( / )

```
export default { base: "/" }
```

Vendor app

```
export default { base: "/vendor/" }
```

Driver app

```
export default { base: "/driver/" }
```

Zone app

```
export default { base: "/zone/" }
```

## Admin app

```
export default { base: "/admin/" }
```

## 2 Build all frontends

Run inside each frontend:

```
npm run build
```

You will get:

```
dist/
```

Move each `dist` into:

```
frontend/{role}/dist
```

## ◆ STEP 2: MODIFY BACKEND TO SERVE ALL FRONTENDS

### app.js (IMPORTANT PART)

Add BEFORE 404 handler:

```
const path = require("path"); // ===== FRONTEND PATHS ===== const userApp =
path.join(__dirname, "../frontend/user/dist"); const vendorApp = path.join(__dirname,
"../frontend/vendor/dist"); const driverApp = path.join(__dirname,
"../frontend/driver/dist"); const zoneApp = path.join(__dirname,
"../frontend/zone/dist"); const adminApp = path.join(__dirname,
"../frontend/admin/dist"); // ===== STATIC FILES =====
app.use(express.static(userApp)); app.use("/vendor", express.static(vendorApp));
app.use("/driver", express.static(driverApp)); app.use("/zone",
express.static(zoneApp)); app.use("/admin", express.static(adminApp)); // ===== SPA
FALLBACKS ===== app.get("/vendor/*", (_, res) => res.sendFile(path.join(vendorApp,
"index.html")) ); app.get("/driver/*", (_, res) => res.sendFile(path.join(driverApp,
"index.html")) ); app.get("/zone/*", (_, res) => res.sendFile(path.join(zoneApp,
"index.html")) ); app.get("/admin/*", (_, res) => res.sendFile(path.join(adminApp,
"index.html")) ); // MAIN APP (ALWAYS LAST) app.get("*", (_, res) =>
res.sendFile(path.join(userApp, "index.html")) );
```

✦ This is the SPA routing fix

✦ This removes all 404 issues

## ◆ STEP 3: COMPILE BACKEND INTO ONE FILE

You want:

```
node server.js
```

✓ Use `@vercel/ncc`

---

### 1 Install NCC

```
npm install -D @vercel/ncc
```

---

### 2 Update `package.json`

```
{ "scripts": { "build": "ncc build server.js -o build", "start": "node server.js" } }
```

---

### 3 Include assets (frontend + uploads)

Create `ncc.config.js`:

```
module.exports = { assets: [ "frontend/**", "public/**", "swagger-output.json" ] };
```

---

### 4 Build backend

```
npm run build
```

Output:

```
build/  
├─ index.js  
└─ assets/
```

---

### 5 Finalize

```
mv build/index.js server.js
```

Now you can **delete source backend folders**:

```
routes/  
middlewares/  
utils/  
config/  
node_modules/
```

(keep backup separately)

## ◆ STEP 4: FINAL DEPLOYMENT (WHAT USER DOES)

User uploads:

```
server.js  
frontend/  
public/  
.env  
package.json
```

Then runs:

```
node server.js
```

OR

```
npm run start
```

- ✓ No NGINX edits
- ✓ No Apache edits
- ✓ No # in URL
- ✓ Clean routes
- ✓ One port
- ✓ One process

## FINAL URL MAP

URL	App
/	User
/vendor	Vendor

URL	App
<code>/driver</code>	Driver
<code>/zone</code>	Zone Manager
<code>/admin</code>	Admin
<code>/api/v1/*</code>	Backend APIs

---

## SECURITY NOTES (IMPORTANT)

- `.env` stays external
- Disable install routes after setup
- Protect `/admin` with auth
- Optional: license check / domain binding

---

## WHY THIS IS THE BEST POSSIBLE SETUP

- SaaS-grade architecture
- Self-host friendly
- aaPanel friendly
- Zero server config dependency
- Easy for **non-technical users**
- Scales to more roles easily

---

## FINAL CONFIRMATION

- ✓ One project out of 6
- ✓ Backend compiled
- ✓ 5 Vite frontends
- ✓ Clean URLs
- ✓ Single deployment
- ✓ `node server.js`