

## D2C OPs — cPanel Deployment Plan 3

**Date:** April 2026 **Deployment model:** PHP 8 backend (php-api/) + Vite-built React frontend on standard cPanel shared hosting. No Node.js required.

**PRIVATE — do not commit this file to a public repository.**

---

### Architecture

Layer	Technology	Where on cPanel
Frontend	React + Vite (static build)	public_html/
Backend API	PHP 8 (php-api/)	public_html/api/
Database	MySQL	cPanel → MySQL Databases
Email	Gmail SMTP via App Password	php-api/.env
Escalation engine	PHP cron (escalation_engine.php)	cPanel → Cron Jobs

---

### User credentials (all 33 accounts)

#### Admin account

Email	Password	Access
admin@d2ctelcare.com	Admin@D2C2026	Full access — all 8 processes

---

#### Password reference by role

Role	Password
Admin (per-process)	Admin@D2C2026
Manager	Manager@123
Operator	Operator@123
Viewer	Viewer@123
Client	Client@123

---

## Ignite

Role	Email	Password
Operator	ops.ignite@d2ctelcare.com	Operator@123
Manager	mgr.ignite@d2ctelcare.com	Manager@123
Viewer	view.ignite@d2ctelcare.com	Viewer@123
Client	client.ignite@d2ctelcare.com	Client@123

## Sunking

Role	Email	Password
Operator	ops.sunking@d2ctelcare.com	Operator@123
Manager	mgr.sunking@d2ctelcare.com	Manager@123
Viewer	view.sunking@d2ctelcare.com	Viewer@123
Client	client.sunking@d2ctelcare.com	Client@123

## RDG

Role	Email	Password
Operator	ops.rdg@d2ctelcare.com	Operator@123
Manager	mgr.rdg@d2ctelcare.com	Manager@123
Viewer	view.rdg@d2ctelcare.com	Viewer@123
Client	client.rdg@d2ctelcare.com	Client@123

## MobiHive

Role	Email	Password
Operator	ops.mobihive@d2ctelcare.com	Operator@123
Manager	mgr.mobihive@d2ctelcare.com	Manager@123
Viewer	view.mobihive@d2ctelcare.com	Viewer@123
Client	client.mobihive@d2ctelcare.com	Client@123

## Momo

Role	Email	Password
Operator	ops.momo@d2ctelcare.com	Operator@123
Manager	mgr.momo@d2ctelcare.com	Manager@123
Viewer	view.momo@d2ctelcare.com	Viewer@123
Client	client.momo@d2ctelcare.com	Client@123

## Sanlam

Role	Email	Password
Operator	ops.sanlam@d2ctelcare.com	Operator@123
Manager	mgr.sanlam@d2ctelcare.com	Manager@123
Viewer	view.sanlam@d2ctelcare.com	Viewer@123
Client	client.sanlam@d2ctelcare.com	Client@123

## MBA

Role	Email	Password
Operator	ops.mba@d2ctelcare.com	Operator@123
Manager	mgr.mba@d2ctelcare.com	Manager@123
Viewer	view.mba@d2ctelcare.com	Viewer@123
Client	client.mba@d2ctelcare.com	Client@123

## Muzanu

Role	Email	Password
Operator	ops.muzanu@d2ctelcare.com	Operator@123
Manager	mgr.muzanu@d2ctelcare.com	Manager@123
Viewer	view.muzanu@d2ctelcare.com	Viewer@123
Client	client.muzanu@d2ctelcare.com	Client@123

Total: 33 accounts — 1 super admin + 4 per process × 8 processes.  
All passwords are case-sensitive.

---

## System credentials

### SMTP — Gmail

Field	Value
SMTP_HOST	smtp.gmail.com
SMTP_PORT	587
SMTP_USER	amosmatimba7@gmail.com
SMTP_PASS	Gmail App Password — Google Account → Security → 2-Step Verification → App passwords → “D2C OPs”

## JWT & engine token

---

Field	How to generate
JWT_SECRET	<code>openssl rand -hex 32</code> — run this once and save the output
ENGINE_TOKEN	<code>openssl rand -hex 32</code> — run this once and save the output

---

## OpenAI

---

Field	Value
OPENAI_API_KEY	Copy the <code>sk-svcacct-...</code> key from <code>server/.env</code>

---

## php-api/.env — fill this before uploading

```
DB_DSN=mysql:host=localhost;dbname=REPLACE_WITH_CPANEL_PREFIXED_DBNAME;charset=utf8mb4
DB_USER=REPLACE_WITH_CPANEL_PREFIXED_DBUSER
DB_PASS=REPLACE_WITH_DB_PASSWORD
JWT_SECRET=REPLACE_WITH_OPENSSL_OUTPUT
CLIENT_URL=https://ops.yourdomain.com
SMTP_HOST=smtp.gmail.com
SMTP_PORT=587
SMTP_USER=amosmatimba7@gmail.com
SMTP_PASS=REPLACE_WITH_GMAIL_APP_PASSWORD
OPENAI_API_KEY=REPLACE_WITH_KEY_FROM_SERVER_ENV
ENGINE_TOKEN=REPLACE_WITH_OPENSSL_OUTPUT
```

---

## Files ready to upload

---

File	Action
<code>public_html.htaccess</code>	Rename to <code>.htaccess</code> , upload to <code>public_html/</code> root
<code>php-api/.htaccess</code>	Upload to <code>public_html/api/</code> — blocks <code>.env</code> access
<code>php-api/.user.ini</code>	Upload to <code>public_html/api/</code> — disables PHP error display

---

## Deployment steps

### Step 1 — Build the frontend locally

```
cd client
npm install
npm run build
```

Output: `client/dist/` — this folder goes into `public_html/`.

### Step 2 — Create the MySQL database in cPanel

1. cPanel → **MySQL Databases**
2. Create database → name it `d2c_ops` (cPanel prefixes it, e.g. `cpuser_d2c_ops`)
3. Create user → set a strong password → note the prefixed username
4. Add user to database → **All Privileges**
5. Fill `DB_DSN`, `DB_USER`, `DB_PASS` in `php-api/.env` with the exact prefixed values cPanel shows you

### Step 3 — Fill in `php-api/.env`

Using the system credentials table above, replace every `REPLACE_WITH_...` placeholder before uploading.

### Step 4 — Upload the frontend

Upload the entire contents of `client/dist/` into `public_html/`:

- cPanel **File Manager** → compress `dist/` into a zip locally → upload → extract in place, **or**
- FTP/SFTP

`public_html/` must contain `index.html` and `assets/` at its root after extraction.

### Step 5 — Upload the PHP API

Upload `php-api/` contents into `public_html/api/`:

```
php-api/index.php      → public_html/api/index.php
php-api/.htaccess      → public_html/api/.htaccess
php-api/.user.ini      → public_html/api/.user.ini
php-api/src/           → public_html/api/src/
php-api/.env           → public_html/api/.env           ← the filled-in file only
```

Do **not** upload `.env.example`.

### Step 6 — Upload the root `.htaccess`

Take `public_html.htaccess` from the project root, **rename it to `.htaccess`**, and upload it to `public_html/` (root level — not inside `api/`).

This file routes `/api/*` to PHP and all other paths to `index.html` so React Router deep-links work.

### Step 7 — Set PHP version in cPanel

1. cPanel → **MultiPHP Manager**
2. Set `public_html/api/` to **PHP 8.0** or later
3. Confirm these extensions are enabled: `pdo`, `pdo_mysql`, `json`, `mbstring`, `openssl`

### Step 8 — Run database migrations

Via SSH:

```
ssh your-cpanel-user@yourdomain.com
export DATABASE_URL="mysql://CPANEL_DB_USER:CPANEL_DB_PASS@localhost:3306/CPANEL_DB_NAME"
cd ~/database
npx knex --knexfile knexfile.js migrate:latest
```

This runs the 5 migration files in `database/migrations/` and creates all tables.

**Do not run seeds unless starting fresh** — seeds wipe all existing data and recreate the 33 accounts from scratch.

To seed a fresh database:

```
npx knex --knexfile knexfile.js seed:run
```

### Step 9 — Set up the escalation engine cron

cPanel → **Cron Jobs** → add:

```
* * * * * php /home/CPANEL_USER/public_html/api/src/escalation_engine.php >> /home/CPANEL_US
```

Replace `CPANEL_USER` with your actual cPanel username. This runs the engine every minute.

To test it manually via HTTP:

```
curl -X POST https://ops.yourdomain.com/api/escalation-engine/run \
  -H "Authorization: Bearer YOUR_ENGINE_TOKEN"
```

---

## Validation checklist

Run these in order after deployment:

#	Check	Expected result
1	GET <code>https://ops.yourdomain.com/api/health</code>	<code>{"status":"ok","timestamp":"..."}</code>

#	Check	Expected result
2	GET <code>https://ops.yourdomain.com</code>	Login page renders
3	Login as <code>admin@d2ctelcare.com</code> <code>/ Admin@D2C2026</code>	Dashboard loads, all 8 processes visible
4	Login as <code>ops.ignite@d2ctelcare.com</code> <code>/ Operator@123</code>	Dashboard loads, only ignite data visible
5	Login as <code>mgr.sunking@d2ctelcare.com</code> <code>/ Manager@123</code>	Dashboard loads, only Sunking data visible
6	Navigate to <code>/incidents</code> then hard-refresh	Page still loads (SPA fallback working)
7	GET <code>https://ops.yourdomain.com/api/.env</code>	Returns <b>403</b>
8	Trigger a notification from the UI	Email arrives at <code>amosmatimba7@gmail.com</code>
9	Create a Sev 1 incident, wait for cron cycle	Tier 1 escalation email delivered
10	Check <code>escalation_executions</code> table	One <b>sent</b> row per fired tier, no duplicates

---

## Cutover checklist

- SSL certificate active on domain (Let's Encrypt via cPanel)
- `php-api/.env` has no `REPLACE_WITH_...` placeholders remaining
- `JWT_SECRET` and `ENGINE_TOKEN` are freshly generated random values
- `GET /api/.env` returns 403
- Gmail App Password tested — email received
- Cron job confirmed running — `/home/CPANEL_USER/logs/escalation.log` is being written
- Database backup taken before going live
- All 33 accounts tested — correct role restrictions apply

---

## Rollback plan

1. Restore previous `public_html/` from a local zip backup (keep one before every upload)
2. If a migration failed, restore MySQL from the pre-migration backup

3. Re-test `/api/health` and the admin login

---

### What does NOT need to happen

- No Node.js app setup in cPanel
- No `npm install` on the server
- No Passenger / PM2 / process manager
- The `server/` directory (Express/TypeScript) is **not deployed** — PHP replaces it entirely