



Windows Setup Guide - Tour Operator System



Current Situation

You have the files downloaded but need to organize them properly for Windows. Here are **3 options** to get your system running:



Option 1: PowerShell Setup (Recommended for Windows)

Step 1: Open PowerShell as Administrator

```
powershell
```

Press Windows + X, then click "Windows PowerShell (Admin)"

Step 2: Create the Complete Setup Script

powershell

Navigate to your folder

```
cd "C:\Users\ACER\tour-operator-system"
```

Create the setup script

```
@"
```

```
# PowerShell Tour Operator Setup Script
```

```
Write-Host "🚀 Tour Operator System - Windows Setup" -ForegroundColor Blue
```

```
Write-Host "=====
```

```
# Check Docker
```

```
if (Get-Command docker -ErrorAction SilentlyContinue) {
```

```
    Write-Host "✅ Docker found" -ForegroundColor Green
```

```
} else {
```

```
    Write-Host "❌ Docker not found. Please install Docker Desktop first:" -ForegroundColor Red
```

```
    Write-Host "    https://www.docker.com/products/docker-desktop/"
```

```
    exit 1
```

```
}
```

```
# Create project structure
```

```
Write-Host "📁 Creating project structure..."
```

```
New-Item -ItemType Directory -Force -Path "backend"
```

```
New-Item -ItemType Directory -Force -Path "frontend\src\components"
```

```
New-Item -ItemType Directory -Force -Path "frontend\public"
```

```
New-Item -ItemType Directory -Force -Path "database"
```

```
# Create docker-compose.yml
```

```
Write-Host "🐳 Creating Docker configuration..."
```

```
@'
```

```
version: '3.8'
```

```
services:
```

```
  frontend:
```

```
    build:
```

```
      context: ./frontend
```

```
    ports:
```

```
      - "3000:3000"
```

```
    environment:
```

```
      - REACT_APP_API_URL=http://localhost:5000/api
```

```
    volumes:
```

```
      - ./frontend:/app
```

```
      - /app/node_modules
```

```
    depends_on:
```

```
      - backend
```

```
  backend:
```

```
    build:
```

```
context: ./backend
ports:
  - "5000:5000"
environment:
  - NODE_ENV=development
  - AMADEUS_CLIENT_ID=Bd76Zxmr3DtsAgSCNVhR1gCzzFDR0M07
  - AMADEUS_CLIENT_SECRET=Onw33473vAI1CTHS
  - AMADEUS_HOSTNAME=test
  - DATABASE_URL=postgresql://tour_operator:secure_password@postgres:5432/tour_operator_db
  - REDIS_HOST=redis
  - REDIS_PORT=6379
  - CORS_ORIGIN=http://localhost:3000
volumes:
  - ./frontend:/app
  - /app/node_modules
depends_on:
  - postgres
  - redis
```

```
postgres:
  image: postgres:15-alpine
  environment:
    - POSTGRES_DB=tour_operator_db
    - POSTGRES_USER=tour_operator
    - POSTGRES_PASSWORD=secure_password
  ports:
    - "5432:5432"
  volumes:
    - postgres_data:/var/lib/postgresql/data
    - ./database/init.sql:/docker-entrypoint-initdb.d/init.sql
```

```
redis:
  image: redis:7-alpine
  ports:
    - "6379:6379"
  volumes:
    - redis_data:/data
```

```
pgadmin:
  image: dpage/pgadmin4:latest
  environment:
    - PGADMIN_DEFAULT_EMAIL=admin@touroperator.com
    - PGADMIN_DEFAULT_PASSWORD=admin123
  ports:
    - "8080:80"
  depends_on:
    - postgres
```

```
volumes:
  postgres_data:
  redis_data:
'@ | Out-File -FilePath "docker-compose.yml" -Encoding UTF8
```

```
Write-Host "✅ Created docker-compose.yml"
```

```
# Create backend files
```

```
Write-Host "🔧 Creating backend files..."
```

```
@'
{
  "name": "tour-operator-backend",
  "version": "1.0.0",
  "main": "server.js",
  "scripts": {
    "start": "node server.js",
    "dev": "nodemon server.js"
  },
  "dependencies": {
    "express": "^4.18.2",
    "amadeus": "^8.1.0",
    "cors": "^2.8.5",
    "dotenv": "^16.3.1",
    "redis": "^4.6.8",
    "pg": "^8.11.3",
    "helmet": "^7.0.0",
    "compression": "^1.7.4",
    "morgan": "^1.10.0"
  },
  "devDependencies": {
    "nodemon": "^3.0.1"
  }
}
'@ | Out-File -FilePath "backend\package.json" -Encoding UTF8
```

```
@'
FROM node:18-alpine
WORKDIR /app
COPY package*.json ./
RUN npm ci
COPY . .
RUN addgroup -g 1001 -S nodejs && adduser -S nodejs -u 1001
RUN chown -R nodejs:nodejs /app
USER nodejs
EXPOSE 5000
CMD ["npm", "run", "dev"]
'
```

```
'@ | Out-File -FilePath "backend\Dockerfile" -Encoding UTF8
```

```
# Create frontend files
```

```
Write-Host "🚧 Creating frontend files..."
```

```
@'
```

```
{  
  "name": "tour-operator-frontend",  
  "version": "1.0.0",  
  "private": true,  
  "dependencies": {  
    "react": "^18.2.0",  
    "react-dom": "^18.2.0",  
    "react-scripts": "5.0.1",  
    "leaflet": "^1.9.4",  
    "axios": "^1.5.0",  
    "lucide-react": "^0.263.1"  
  },  
  "scripts": {  
    "start": "react-scripts start",  
    "build": "react-scripts build"  
  },  
  "devDependencies": {  
    "tailwindcss": "^3.3.3",  
    "autoprefixer": "^10.4.15",  
    "postcss": "^8.4.28"  
  }  
}
```

```
'@ | Out-File -FilePath "frontend\package.json" -Encoding UTF8
```

```
@'
```

```
FROM node:18-alpine  
WORKDIR /app  
COPY package*.json ./  
RUN npm ci  
COPY . .  
EXPOSE 3000  
CMD ["npm", "start"]
```

```
'@ | Out-File -FilePath "frontend\Dockerfile" -Encoding UTF8
```

```
# Create database init
```

```
Write-Host "🗄️ Creating database schema..."
```

```
@'
```

```
CREATE TABLE IF NOT EXISTS locations (  
  id SERIAL PRIMARY KEY,  
  iata_code VARCHAR(3) UNIQUE,  
  name VARCHAR(255) NOT NULL,  
  city VARCHAR(100),
```

```

country VARCHAR(100),
latitude DECIMAL(10,8),
longitude DECIMAL(11,8)
);

INSERT INTO locations (iata_code, name, city, country, latitude, longitude) VALUES
('MAD', 'Madrid-Barajas Airport', 'Madrid', 'Spain', 40.472219, -3.560833),
('BCN', 'Barcelona-El Prat Airport', 'Barcelona', 'Spain', 41.2971, 2.0785),
('LHR', 'Heathrow Airport', 'London', 'United Kingdom', 51.4706, -0.4619),
('CDG', 'Charles de Gaulle Airport', 'Paris', 'France', 49.0097, 2.5479),
('BKK', 'Suvarnabhumi Airport', 'Bangkok', 'Thailand', 13.6900, 100.7501)
ON CONFLICT (iata_code) DO NOTHING;
'@ | Out-File -FilePath "database\init.sql" -Encoding UTF8

```

```

Write-Host "🚀 Starting Docker containers..."
docker-compose up -d

```

```

Write-Host ""
Write-Host "🌐 Setup complete!" -ForegroundColor Green
Write-Host "🌐 Frontend: http://localhost:3000" -ForegroundColor Yellow
Write-Host "🔑 Backend: http://localhost:5000/api" -ForegroundColor Yellow
Write-Host "🗄 Database: http://localhost:8080" -ForegroundColor Yellow
"@ | Out-File -FilePath "setup-windows.ps1" -Encoding UTF8

```

```

# Run the setup
Write-Host "Running setup script..."
powershell -ExecutionPolicy Bypass -File "setup-windows.ps1"

```

🚀 Option 2: Manual File Creation (Step by Step)

Step 1: Organize Your Files

bash

In your tour-operator-system folder, create this structure:

```
tour-operator-system/  
├─ docker-compose.yml  
├─ backend/  
│   ├─ package.json  
│   ├─ Dockerfile  
│   └─ server.js  
├─ frontend/  
│   ├─ package.json  
│   ├─ Dockerfile  
│   └─ src/  
└─ database/  
    └─ init.sql
```

Step 2: Create docker-compose.yml

Create a new file `docker-compose.yml` and copy this content:

version: '3.8'

services:

frontend:

build:

context: ./frontend

ports:

- "3000:3000"

environment:

- REACT_APP_API_URL=http://localhost:5000/api

volumes:

- ./frontend:/app
- /app/node_modules

depends_on:

- backend

backend:

build:

context: ./backend

ports:

- "5000:5000"

environment:

- NODE_ENV=development
- AMADEUS_CLIENT_ID=Bd76Zxmr3DtsAgSCNVhRlgCzzFDR0M07
- AMADEUS_CLIENT_SECRET=Onw33473vAI1CTHS
- AMADEUS_HOSTNAME=test
- DATABASE_URL=postgresql://tour_operator:secure_password@postgres:5432/tour_operator_db
- REDIS_HOST=redis
- REDIS_PORT=6379
- CORS_ORIGIN=http://localhost:3000

volumes:

- ./backend:/app
- /app/node_modules

depends_on:

- postgres
- redis

postgres:

image: postgres:15-alpine

environment:

- POSTGRES_DB=tour_operator_db
- POSTGRES_USER=tour_operator
- POSTGRES_PASSWORD=secure_password

ports:

- "5432:5432"

volumes:

- postgres_data:/var/lib/postgresql/data
- ./database/init.sql:/docker-entrypoint-initdb.d/init.sql

redis:

image: redis:7-alpine

ports:

- "6379:6379"

volumes:

- redis_data:/data

pgadmin:

image: dpage/pgadmin4:latest

environment:

- PGADMIN_DEFAULT_EMAIL=admin@touroperator.com
- PGADMIN_DEFAULT_PASSWORD=admin123

ports:

- "8080:80"

depends_on:

- postgres

volumes:

postgres_data:

redis_data:

Step 3: Start the System

bash

In PowerShell or Command Prompt:

cd "C:\Users\ACER\tour-operator-system"

docker-compose up -d

Option 3: Use WSL2 (Windows Subsystem for Linux)

Step 1: Install WSL2

powershell

In PowerShell as Administrator:

wsl --install

Restart your computer when prompted

Step 2: Use Linux Commands

```
bash
```

```
# Open Ubuntu from Start Menu, then:
```

```
cd /mnt/c/Users/ACER/tour-operator-system
```

```
bash complete_auto_setup.sh
```

Quick Fix - Using Your Current Files

Since you already have `complete_auto_setup.sh`, try this:

Using Git Bash (if installed):

```
bash
```

```
# Right-click in your folder → "Git Bash Here"
```

```
chmod +x complete_auto_setup.sh
```

```
./complete_auto_setup.sh
```

Using Docker Desktop Terminal:

```
bash
```

```
# Open Docker Desktop → Click "Terminal" button
```

```
cd /host_mnt/c/Users/ACER/tour-operator-system
```

```
chmod +x complete_auto_setup.sh
```

```
./complete_auto_setup.sh
```

Prerequisites Check

Before running, make sure you have:

✓ Docker Desktop for Windows

- Download: <https://www.docker.com/products/docker-desktop/>
- Make sure it's running (whale icon in system tray)

✓ Ports Available

```
powershell
```

```
# Check if ports are free:
```

```
netstat -an | findstr ":3000"
```

```
netstat -an | findstr ":5000"
```

```
# Should return nothing if ports are free
```

Simplified One-File Solution

Create a new file called `start-system.bat` with this content:

```
batch

@echo off
echo 🚀 Starting Tour Operator System...

:: Check Docker
docker --version >nul 2>&1
if errorlevel 1 (
    echo ❌ Docker not found! Please install Docker Desktop
    echo https://www.docker.com/products/docker-desktop/
    pause
    exit /b 1
)

echo ✅ Docker found

:: Create folders
mkdir backend 2>nul
mkdir frontend\src\components 2>nul
mkdir frontend\public 2>nul
mkdir database 2>nul

:: Start Docker Compose
echo 🚀 Starting services...
docker-compose up -d

echo.
echo 🎉 System starting!
echo 🌐 Frontend: http://localhost:3000
echo 🕒 Backend: http://localhost:5000/api
echo 🗄 Database: http://localhost:8080
echo.
pause
```

Then just **double-click** `start-system.bat` to run!

Still Having Issues?

Tell me:

1. Do you have **Docker Desktop** installed and running?
2. What happens when you run: `docker --version` in Command Prompt?

3. Which method would you prefer to try first?

I'll help you get it working! 🚀