Demo 4: Technical Installation Manual

GreenCart
Client: BBD Software

Team member	Student Number
Nikhil Govender	u22504193
Shayden Naidoo	u23599236
Corné de Lange	u23788862
Tshegofatso Mahlase	u22580833
Samvit Prakash	u23525119

1) Introduction

GreenCart is a sustainability-focused e-commerce platform comprising a **FastAPI** backend (Python 3.11), a **React + Vite** frontend, and a **PostgreSQL** database. Optional integrations include **AWS S3** for image storage. This manual explains how to install all prerequisites, configure environment variables, run the system locally on Windows/Linux/macOS, and (optionally) run via Docker.

Repository layout (key paths)

Green-Cart/

app/ # FastAPI backend source

frontend/ # React + Vite frontend

tests/ unit-tests/ integration-tests/ # Python tests

requirements.txt # Backend pinned deps (UTF-16 encoded)

Dockerfile # Backend container build

.env # Example/prod env (do NOT commit credentials)

documents/, docs/ # Diagrams, assets

uploads/ # Local static uploads (optional)

Ports (default): Backend `8000`, Frontend `5173`.

---

2) Prerequisites

2.1 Computer Hardware Requirements

Resource	Minimum	Recommended
CPU	4 cores	8+ cores
RAM	8 GB	16 GB
Disk	10 GB free	20+ GB SSD

2.2 Software Applications

Install these before proceeding:

Software	Version	Purpose
Git	≥ 2.40	Clone repo
Python 3.11.x		Backend runtime
pip	≥ 23.x	Python packages
Node.js 20.x LTS		Frontend tooling
npm	≥ 10.x	Frontend packages
PostgreSQL	15.x	Database
Docker *(optional)*	Latest	Container build/run

Tip: Use **pyenv** (Linux/macOS) or the official **Python 3.11** installer (Windows). On macOS, prefer **Homebrew**.

2.3 Major Packages (pinned)

Backend (from `requirements.txt`):

- fastapi **0.115.12**, uvicorn **0.34.2**, starlette **0.46.2**
- SQLAlchemy **2.0.41**, psycopg2-binary **2.9.10**, python-dotenv **1.1.0**
- pydantic **2.11.5**, bcrypt **4.3.0**, python-jose **3.4.0**
- boto3 **1.35.98** *(only if using S3)*

Frontend (from `frontend/package.json`):

- vite **^6.x**, react, axios, highcharts, vitest, @testing-library/*

---

3) Services & Accounts

3.1 Database

- **Local PostgreSQL** (recommended for development), or
- **Supabase / RDS** (team-managed). Obtain a **dev** connection string from the team if you don't want to set up local Postgres.

3.2 AWS S3 (optional)

Needed only if you want cloud image uploads via `/images/upload`:

- `AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, `AWS_REGION`, `AWS_S3_BUCKET_NAME`
- Optional: `S3_BASE_URL` (if you use a custom CDN domain)

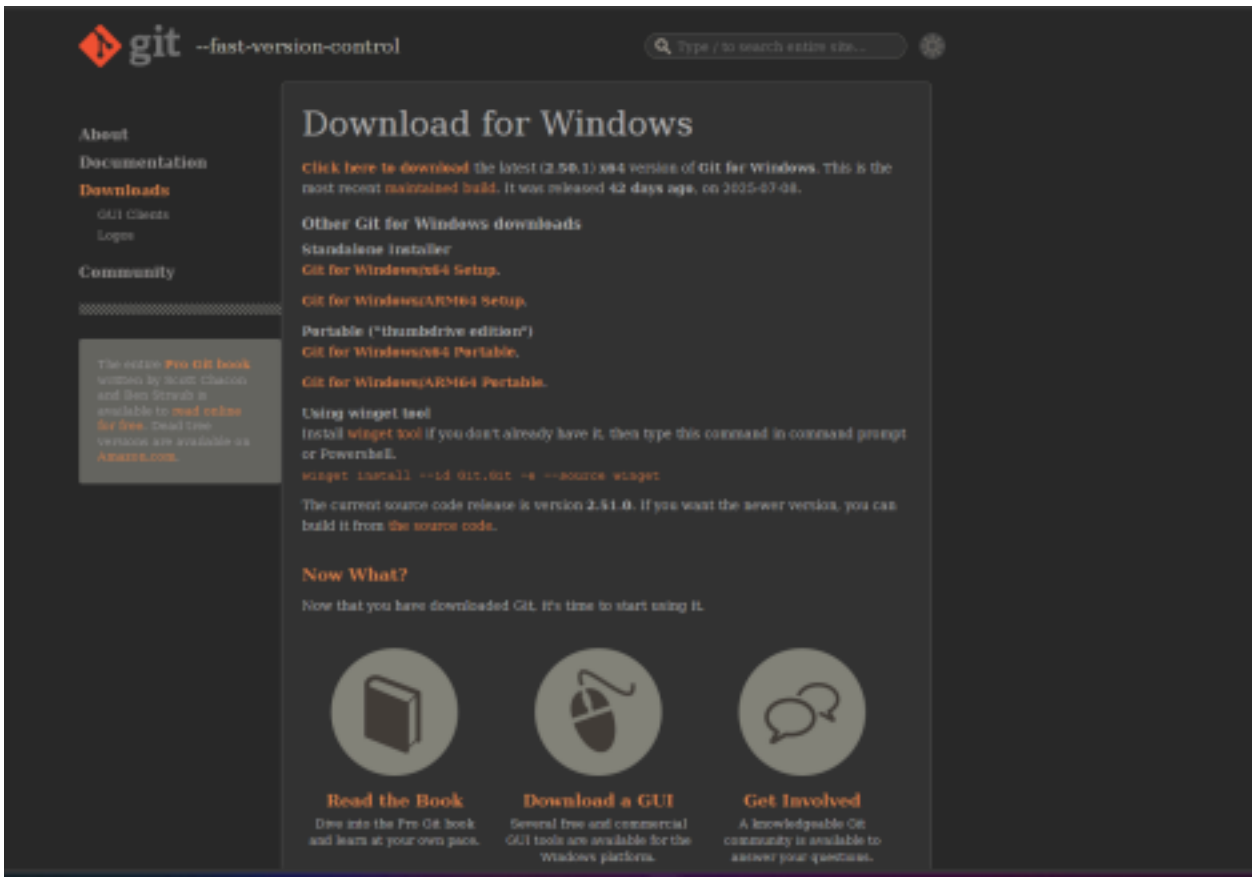
Without S3 vars, the API will run, but S3 endpoints will be disabled.

---

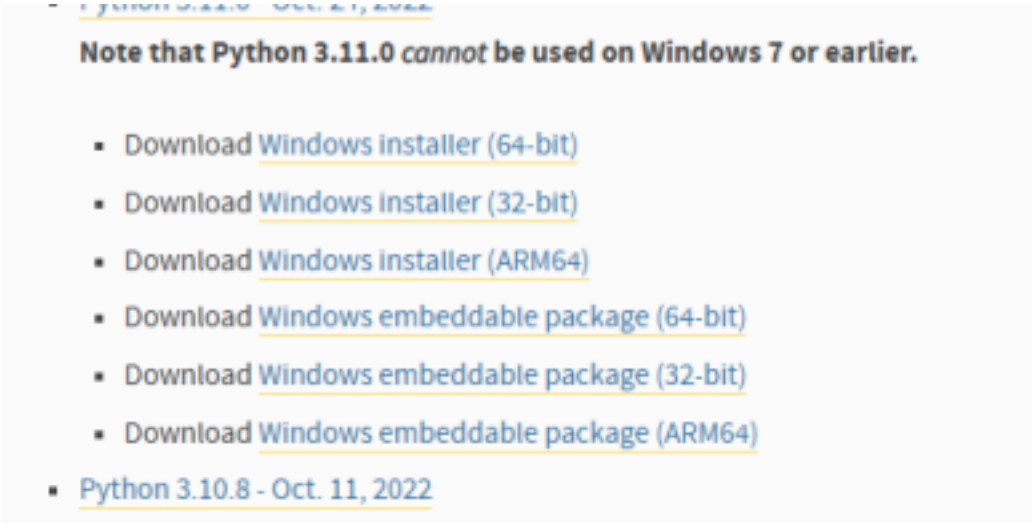
4) OS-Specific Setup

4.1 Windows 10/11

1. **Git**: Install from git-scm.com or Microsoft Store.



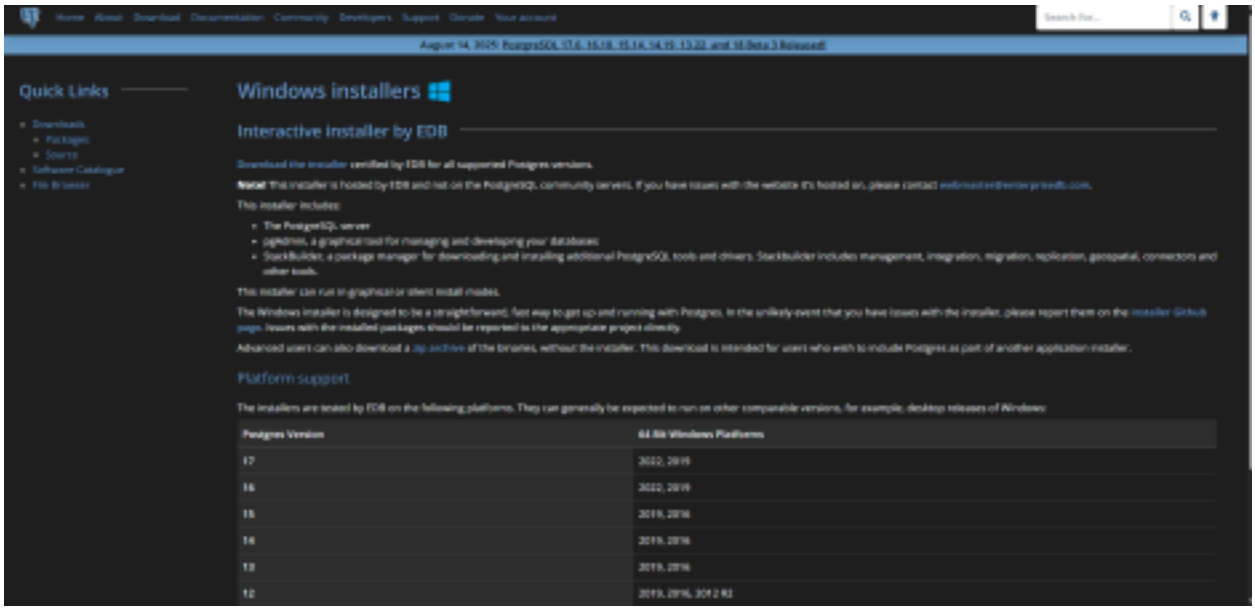
2. **Python 3.11**: Use the official Windows installer; check `python --version`.



3. **Node.js 20 LTS**: Download from nodejs.org; check `node -v` and `npm -v`.

---

4. **PostgreSQL 15**: Install via EnterpriseDB installer; create a local superuser.



5. *(Optional)* **Docker Desktop**: Enable WSL2 backend.

4.2 Ubuntu 22.04+

sudo apt update

sudo apt install -y git curl build-essential python3.11 python3.11-venv python3-pip postgresql

Node 20 LTS

curl -fsSL https://deb.nodesource.com/setup_20.x | sudo -E bash -

sudo apt install -y nodejs

4.3 macOS (Apple Silicon/Intel)

Homebrew

/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

Tools

brew install git python@3.11 node postgresql

5) Get the Code

git clone https://github.com/COS301-SE-2025/Green-Cart

cd Green-Cart

Ensure you're at the folder that contains `app/`, `frontend/`, `requirements.txt`.

6) Environment Configuration

Create a **project-root** `.env` file (do **not** commit). The backend reads it via `python-dotenv` on startup.

6.1 Minimum required variables

Option A: full connection string

DATABASE_URL=postgresql://<user>:<password>@<host>:5432/<db>

Option B: components (used if DATABASE_URL not set)

DB_HOST=localhost

DB_PORT=5432

DB_USER=postgres

DB_PASS=<password>

DB_NAME=greencart

Optional S3 (enables /images endpoints)

AWS_ACCESS_KEY_ID=

AWS_SECRET_ACCESS_KEY=

AWS_REGION=

AWS_S3_BUCKET_NAME=

Optional CDN base

S3_BASE_URL=

> The repo includes a `.env` file pointing to **production**. **Do not** use or commit production credentials. Overwrite with your local/dev values.

6.2 Frontend API base URL

The frontend central config is at `frontend/src/config/api.js`.

- **Production** default is hard-coded to: `https://api.greencart-cos301.co.za`.
- For **local development**, set it to your local API:

// frontend/src/config/api.js

export const API_BASE_URL = import.meta.env.VITE_API_URL ||

"http://127.0.0.1:8000"; export const getApiUrl = () => API_BASE_URL;

export default API_BASE_URL;

Then create `frontend/.env`:

VITE_API_URL=http://127.0.0.1:8000

> Some older service files import from `config/api.js` (recommended path). If any file still hard-codes the production URL, update it to use `API_BASE_URL`.

---

7) Backend Installation & Run (FastAPI)

7.1 Create virtual environment & install deps

From repo root

python3.11 -m venv venv

Activate

Linux/macOS

```
source venv/bin/activate

# Windows

venv\Scripts\activate


# requirements.txt is UTF-16 encoded — pip handles it fine

pip install -r requirements.txt
```

7.2 Initialize the database (local Postgres)

Create a database and user, then apply schema:

```
# psql

createdb greencart -U postgres

# (Optional) Create a dedicated user and grant privileges
```

> The project uses SQLAlchemy models. If starting from an empty DB and **no migrations** exist, you can bootstrap tables with a one-liner:

```
python -c "import app.models as m; from app.db.database import Base,
engine; Base.metadata.create_all(engine)"
```

7.3 Launch the API

uvicorn app.main:app --reload --port 8000

API docs: http://127.0.0.1:8000/docs

Health: http://127.0.0.1:8000/health

CORS: Allowed origins include `http://localhost:5173` and `http://localhost:3000` by default.

8) Frontend Installation & Run (React + Vite)

cd frontend

npm install

Ensure VITE_API_URL is set as per §6.2 for local dev

npm run dev

Opens at http://localhost:5173

9) Docker (Optional, Backend)

If you prefer containers, a `Dockerfile` is provided for the API.

9.1 Build & run

Build image

docker build -t greencart-api .

Run (pass env file)

docker run --name greencart-api \

--env-file .env \

-p 8000:8000 \

greencart-api

> You can author a simple `docker-compose.yml` with `api` and a `db` service if desired.

---

10) Verifying Your Setup

1. **API health**: `GET http://127.0.0.1:8000/health` should return `{ status: "healthy" }`.
2. **Open frontend**: `http://localhost:5173`.
3. **Authentication**: Use `/signin` (user) or `/admin/signin` (admin). Ensure the frontend points to the same API base.
4. **Products/Cart**: Browse products, add to cart, place orders.

Key API prefixes (from `app/routes/`):

- `GET /health`
- `POST /signin`, `POST /signup`
- `GET /users/{id}`
- `GET /products/*`, `POST /products/sales_metrics`

- `POST /cart/*`, `POST /orders/*`
- `POST /images/upload` *(requires S3 env)*

- `GET/POST /admin/*` (admin/auth/metrics/products)

11) Testing

11.1 Backend (pytest)

From repo root (ensure venv activated)

pytest -q unit-tests

pytest -q integration-tests

11.2 Frontend (Vitest)

cd frontend

npm run test

or

npm run test:run

12) Troubleshooting

Symptom	Likely Cause	Fix
`psycopg2.OperationalError`	Bad `DATABASE_URL` / DB not running	Verify `.env`, start Postgres, confirm network access
Frontend calls hitting prod API	Hard-coded URL in `src/config/api.js`	Apply §6.2 and restart dev server
CORS error in browser	Origin not allowed	Use `http://localhost:5173` or add dev origin in backend CORS list
Windows EPERM with Vite	Path under OneDrive or locked `.vite` cache	Move repo out of OneDrive; `rd /s /q node_modules\.vite`; restart shell as Admin
`ModuleNotFoundError` in tests	Not in venv / PYTHONPATH	Activate venv; run from repo root

	Missing AWS vars Set AWS env in `.env` or skip S3 features
S3 upload endpoints disabled	

13) Deployment Notes (Production)

- **API**: Run `uvicorn` behind Nginx/Gunicorn on a Linux host; set `.env` with production DB & S3. Expose 8000 internally.
- **Frontend**: `npm run build` produces `frontend/dist/` → serve via Nginx or a static host. **Ensure** `API_BASE_URL` points to your production API domain.
- **Security**: Do **not** commit `.env`. Rotate credentials periodically.

14) Using GreenCart

To see how to use the system, you can refer to the User Manual - Google Docs

15) Appendix

A. Directory Tree (Top-Level)

Green-Cart/

app/ frontend/ tests/ unit-tests/ integration-tests/

requirements.txt Dockerfile .env documents/ uploads/

B. Notable Files

- `app/main.py` – FastAPI app, routers & CORS configuration

- `app/db/database.py` – environment-based DB configuration (`DATABASE_URL` or components)

- `app/routes/…` – REST endpoints (products, cart, orders, auth, admin, images)

- `app/services/s3_service.py` – S3 integration (enabled via AWS env vars)
- `frontend/src/config/api.js` – Frontend API base URL