# CRM Application Installation Guide

This guide will walk you through setting up and running the full-stack, containerized CRM application on your local machine.

---

## 1. Prerequisites

Before you begin, ensure you have the following software installed on your computer:

- **Git**: For cloning the source code repository.
- **Docker** and **Docker Compose**: For building and running the application containers. Docker Desktop for Windows and Mac includes both.

---

## 2. Installation Steps

### Step 1: Clone the Repository

First, clone the project's source code from GitHub to your local machine. Open your terminal or command prompt and run the following command:

```
Bash
```

```
git clone https://github.com/your-username/your-crm-repo.git
```

Navigate into the newly created project directory:

```
Bash
```

```
cd your-crm-repo
```

### Step 2: Create the Environment File

The application requires an environment file to store secrets and configuration details.

Create a new file named .env in the **root directory** of the project. Copy and paste the following content into it, replacing the placeholder values with your own.

**File: .env**

Code snippet

```
# PostgreSQL Database Credentials
# Choose a username, password, and database name for your local instance.
# The database container will be automatically created with these settings.
DB_USER=crm_user
DB_PASSWORD=your_secure_password
DB_DATABASE=crm_db
DB_HOST=db
DB_PORT=5432

# Backend Server Port
# This is the port the backend will run on inside Docker and be exposed to on your machine.
PORT=5000

# JSON Web Token Secret
# This should be a long, random, and secret string.
JWT_SECRET=your_super_secret_key_that_is_long_and_random
```

---

## 3. Running the Application

With the configuration in place, you can now build and run the entire application stack with a single command.

### Step 1: Build and Run Containers

In your terminal, from the project's root directory, run:

Bash

```
docker-compose up --build
```

This command will:

1. Pull the necessary base images (PostgreSQL, Node.js).
2. Build your custom backend and frontend images.
3. Start all three containers (db, backend, frontend).
4. On the very first run, it will automatically create the database and run all the SQL scripts (01_schema.sql, 02_master_data.sql, and 03_development_seed.sql) to set up your tables and populate them with data.

Wait for the build process to complete and the logs to stabilize. You will see output from all three services.

### Step 2: Access the Application

Once the containers are running, open your web browser and navigate to: http://localhost:3000

---

## 4. Default Login Credentials

The application is now running. You will be redirected to the login page. Use the following credentials to test the different user roles.

- **Supervisor / Admin (First-Time Setup)**:
  - **Email**: admin@example.com
  - **Password**: Enter any password to begin. You will be redirected to a setup page to create your permanent password.
- **Sales Manager** (can see all data):
  - **Email**: jane.smith@example.com
  - **Password**: password123
- **Sales User** (sees only their own data):
  - **Email**: john.doe@example.com
  - **Password**: password123
- **Product Manager**:
  - **Email**: peter.pan@example.com
  - **Password**: password123

## 5. Stopping the Application

To stop the application, go to the terminal where the containers are running and press **Ctrl + C**.

To stop the containers and completely remove the database (if you want to start fresh again), run:

Bash

```
docker-compose down -v
```