Save-n-Bite Technical Installation Manual

Table of Contents

Introduction

Save-n-Bite is a comprehensive food waste reduction system designed to help users manage their food inventory and reduce waste through intelligent recommendations and community sharing features. The system consists of multiple components that need to be properly installed and configured:

- **Frontend Application**: User interface built with React 18.2.0
- **Backend API**: Server-side application built with Django 5.2.3, Django Rest Framework 3.16.0
- **Database**: PostgreSQL 16.9
- **Additional Services**: Azure Emulator 3.35.0, Python 3.12.3
- This manual provides step-by-step instructions to clone, configure, and run the Save-n-Bite system on your local development environment. The installation process has been tested on Windows 10/11, macOS 12+, and Ubuntu 20.04+ LTS.

Prerequisites

Before installing Save-n-Bite, ensure your system meets the following requirements and has the necessary software installed.

System Requirements

Minimum Hardware Requirements:

- RAM: 8GB (16GB recommended)
- Storage: 10GB free space
- Processor: Dual-core 2.5GHz or equivalent

Supported Operating Systems:

- Windows 10/11
- macOS 12.0 or later
- Ubuntu 20.04 LTS or later
- Other Linux distributions (with manual dependency management)

Required Software

1. Git (Version 2.34.0 or later)

Installation Instructions:

Windows:

1. Download Git from https://git-scm.com/download/win
2. Run the installer and follow the setup wizard

3. Verify installation: Open Command Prompt and run `git --version`

macOS:

```
# Using Homebrew (recommended)
brew install git

# Or download from https://git-scm.com/download/mac
```

Linux (Ubuntu/Debian):

```
sudo apt update
sudo apt install git
```

2. Node.js (Version 18.17.0 or later) and npm

Windows & macOS:

1. Download from https://nodejs.org/
2. Install the LTS version
3. Verify installation:

```
node --version
npm --version
```

Linux (Ubuntu/Debian):

```
curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
sudo apt-get install -y nodejs
```

3. Database System

PostgreSQL 16.x:

Windows:

1. Download from https://www.postgresql.org/download/windows/
2. Run installer and note down the password for the `postgres` user

macOS:

```
brew install postgresql@15
brew services start postgresql@15
```

Linux (Ubuntu/Debian):

```
sudo apt update
sudo apt install postgresql postgresql-contrib
sudo systemctl start postgresql
sudo systemctl enable postgresql
```

4. Additional Dependencies

Docker (Optional but recommended for containerized deployment):

- Download and install Docker Desktop from https://www.docker.com/products/docker-desktop

---

Installation

Step 1: Clone the Repository

1. Open your terminal/command prompt
2. Navigate to your desired directory
3. Clone the repository:

```
git clone https://github.com/COS301-SE-2025/Save-n-Bite.git
cd Save-n-Bite
```

Step 2: Backend Setup

1. Navigate to the backend directory:

```
cd save-n-bite-backend
```

2. Install backend dependencies:

```
pip install -r requirements.txt
```

3. Create environment configuration:

```
source venv/bin/activate
```

4. Edit the `.env` file with your configuration:

```
# Database Configuration
```

```
SECRET_KEY=your_secret_key
DEBUG=False
DB_NAME=save_n_bite_db
DB_USER=your_username
DB_PASSWORD=your_password
DB_HOST=localhost
DB_PORT=5432

# Development (using Azurite emulator)
ENVIRONMENT=development
AZURE_ACCOUNT_NAME=your_account_name
AZURE_ACCOUNT_KEY=your_account_key
AZURE_CONTAINER_NAME=savenbite-media
```

5. Set up the database:

Create database
createdb savenbit_db

Run migrations
Python manage.py migrate

Step 3: Frontend Setup

1. Open a new terminal and navigate to the frontend directory:

cd save-n-bite-frontend

2. Install frontend dependencies:

npm install

3. Create frontend environment configuration:

source venv/bin/activate

4. Configure the frontend environment:

```
$ .env.prod
# Django
SECRET_KEY=your-super-secret-production-key-here
DEBUG=0
ALLOWED_HOSTS=yourdomain.com,www.yourdomain.com,localhost

# Database
POSTGRES_DB=myapp_prod
POSTGRES_USER=myapp_user
POSTGRES_PASSWORD=secure-database-password-here

# Optional: Email settings
EMAIL_HOST=smtp.gmail.com
EMAIL_PORT=587
EMAIL_USE_TLS=1
EMAIL_HOST_USER=your-email@gmail.com
EMAIL_HOST_PASSWORD=your-app-password

# Optional: AWS S3 for static files
AWS_ACCESS_KEY_ID=your-aws-key
AWS_SECRET_ACCESS_KEY=your-aws-secret
AWS_STORAGE_BUCKET_NAME=your-bucket-name
AWS_S3_REGION_NAME=us-east-1
```

Step 4: Verify Installation

1. Check that all dependencies are properly installed:

In backend directory
pip list

```
Package                       Version
----------------------------- -----------
amqp                          5.3.1
asgiref                       3.8.1
azure-core                    1.35.0
azure-identity                1.24.0
azure-storage-blob            12.26.0
billiard                      4.2.1
celery                        5.5.3
certifi                       2025.8.3
cffi                          1.17.1
charset-normalizer            3.4.3
click                         8.2.1
click-didyoumean              0.3.1
click-plugins                 1.1.1.2
click-repl                    0.3.0
contourpy                     1.3.2
coverage                      7.9.1
cryptography                  45.0.6
cycler                        0.12.1
dj-database-url               3.0.0
Django                        5.2.3
django-cors-headers           4.7.0
django-redis                  6.0.0
djangorestframework           3.16.0
djangorestframework_simplejwt 5.5.0
fonttools                     4.58.4
idna                          3.10
iniconfig                     2.1.0
isodate                       0.7.2
joblib                        1.5.1
kiwisolver                    1.4.8
kombu                         5.5.4
matplotlib                    3.10.3
model-bakery                  1.20.5
msal                          1.33.0
msal-extensions               1.3.1
numpy                         2.3.1
packaging                     25.0
pandas                        2.3.0
pillow                        11.2.1
pip                           24.0
pluggy                        1.6.0
```

In frontend directory
npm list

```
(venv) somworld@DESKTOP-MBUGLIA:~/COS301/Capstone/Save-n-Bite/save-n-bite-frontend$ npm list
save-n-bite-frontend@0.1.0 /home/somworld/COS301/Capstone/Save-n-Bite/save-n-bite-frontend
├── @babel/core@7.28.3
├── @babel/preset-env@7.28.0
├── @babel/preset-react@7.27.1
├── @testing-library/jest-dom@5.17.0
├── @testing-library/react@13.4.0
├── @testing-library/user-event@14.6.1
├── @types/node@22.16.2
├── @types/react-dom@19.1.6
├── @types/react@19.1.8
├── @vitejs/plugin-react@5.0.1
├── autoprefixer@10.4.21
├── axios@1.10.0
├── babel-jest@27.5.1
├── esbuild@0.25.6
├── framer-motion@12.23.0
├── identity-obj-proxy@3.0.0
├── jest-environment-jsdom@27.5.1
├── jest-transform-stub@2.0.0
├── jest@27.5.1
├── leaflet@1.9.4
├── lucide-react@0.511.0
├── postcss@8.5.6
├── prop-types@15.8.1
├── react-dom@18.3.1
├── react-hot-toast@2.6.0
├── react-is@18.3.1
├── react-leaflet@4.2.1
├── react-router-dom@6.30.1
├── react@18.3.1
├── recharts@3.1.0
├── sonner@2.0.6
├── tailwindcss@3.4.17
└── vite@7.1.3
```

2. Verify database connection:

In backend directory
psql -h localhost -U save_user -d save_n_bite_db

```
(venv) somworld@DESKTOP-MBUGLIA:~/COS301/Capstone/Save-n-Bite/save-n-bite-backend$ psql -h localhost -U postgres -d save_n_bite_db
Password for user postgres:
psql (16.9 (Ubuntu 16.9-0ubuntu0.24.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, compression: off)
Type "help" for help.

save_n_bite_db=#
```

Deployment/Running

Development Environment

Option 1: Manual Startup

1. **Start the Backend Server:**

```
# Navigate to backend directory
cd save-n-bite-backend
<Ctrl><Shift>+<P> Azurite: start
python manage.py runserver
```

The backend server should start on `http://localhost:8000`

2. **Start the Frontend Application:**

```
# Open new terminal, navigate to frontend directory
cd save-n-bite-frontend
npm start
```

The frontend application should start on `http://localhost:8001`

Option 2: Docker Deployment

```
# From project root directory
docker-compose up -d
```

Production Environment

1. **Build the Frontend:**

```
cd save-n-bite-frontend
npm run build
```

2. **Start Production Backend:**

```
cd save-n-backend
python manage.py runserver
```

Accessing the Application

Once both servers are running:

- **Frontend**: Open your browser and go to `http://localhost:8001`
- **Backend API**: Available at `http://localhost:3000`

Default Login Credentials (Development)

Email: admin@savenbit.com
Password: admin123

Note: Change these credentials immediately in production environments

---

Troubleshooting

Common Issues

Issue 1: Port Already in Use

Error: `EADDRINUSE: address already in use :::8000`

Solution:

```
# Find process using the port
lsof -i :8000  # macOS/Linux
netstat -ano | findstr :8000  # Windows

# Kill the process
kill -9 <PID>  # macOS/Linux
taskkill /PID <PID> /F  # Windows
```

Issue 2: Database Connection Error

Error: `Connection refused` or `Database does not exist`

Solutions:

1. Ensure PostgreSQL service is running
2. Verify database credentials in `.env`
3. Create the database if it doesn't exist:

```
createdb savenbit_db
```

Issue 3: Module Not Found Errors

Error: `Module 'xyz' not found`

Solution:

```
# Clear npm cache and reinstall
rm -rf node_modules package-lock.json
npm cache clean --force
npm install
```

Issue 4: Permission Errors (Linux/macOS)

Solution:

sudo chown -R $(whoami) ~/.npm

Logs and Debugging

- **Backend Logs**: Check `logs/` directory or console output
- **Frontend Logs**: Check browser developer console
- **Database Logs**: Check PostgreSQL logs in system logs directory

---

Additional Resources

Documentation Links

- **User Manual**: [Link to user manual documentation]
- **API Documentation**: [Link to service contracts]
- **Database Schema**: [Link to database documentation]

Development Tools

- **Postman Collection**: [Link to API testing collection]
- **Database Admin Tools**: pgAdmin

Support and Contribution

- **GitHub Repository**: https://github.com/COS301-SE-2025/Save-n-Bite
- **Issue Tracking**: [GitHub Issues link]

Version Information

- **Current Version**: v1.0.0
- **Node.js**: 18.17.0+
- **npm**: 9.6.7+
- **PostgreSQL**: 15.x

Last Updated: [19 August 2025] Manual Version: 1.0

For technical support or questions regarding this installation manual, please create an issue in the GitHub repository or contact the development team.