

# *Agro Nursery Online Store*

## Документация

**Автор:**

Иван Иванов Петров

**Факультетен номер:**

ЗМІ0700035

## С ъ д ъ р ж а н и е

Описание на проекта	3
Цели на проекта.....	3
Основна функционалност.....	4
Модел на данните.....	6
Основни изгледи.....	9
API ресурси (Сървър).....	10
Използвани технологии	13
Архитектура.....	15
Конфигурация на приложението.....	16
Източници на информация.....	17

## Описание на проекта

**The Agro Nursery Online Store** е уеб-базирана платформа, разработена с цел да улесни клиентите при закупуването на продукти от разсадници. Системата

предоставя достъп до широка гама от растения, семена и градинарски пособия, като осигурява удобен и интуитивен начин за търсене, разглеждане и поръчка на продукти.

## Цели на проекта

### 1. Осигуряване на лесен достъп до продукти

- ✓ Разработване на платформа, която позволява на клиентите бързо и лесно да намират и поръчват необходимите им продукти.
- ✓ Предоставяне на детайлна информация за артикулите, включително изображения, описания и характеристики.

### 2. Осигуряване на надеждна поддръжка и обслужване на клиентите

- ✓ Внедряване на система за бърза и ефективна комуникация с клиентите чрез чат или имейл.
- ✓ Предоставяне на актуална информация за доставки и статус на поръчките.

## Основна функционалност

Проектът **The Agro Nursery Online Store** предоставя широка гама от функционалности, които улесняват потребителите при намирането, закупуването и управлението на продукти от различни разсадници. Платформата осигурява удобен интерфейс за взаимодействие между клиенти, поддръжка и администратори, като предлага следните основни възможности:

### 1. Преглед на информация и продукти

- ✓ Потребителите могат да разглеждат страниците на уебсайта, включително **Начало, Продукти**.
- ✓ Осигурена е възможност за свободно разглеждане на наличните продукти, преди да се извърши покупка.

### 2. Регистрация и управление на потребителски профили

- ✓ **Анонимните потребители** могат да създадат акаунт, като предоставят валиден имейл, име, фамилия и парола. По подразбиране **новите потребители** получават роля **Клиент**.

### 3. Управление на потребителски данни

- ✓ **Регистрираните потребители** могат да преглеждат и редактират личната си информация.
- ✓ **Администраторите** могат да редактират потребителски данни и да им присвояват роли като **Клиент** или **Администратор** в системата, както и да премахват потребители.

#### **4. Управление на продукти**

- ✓ На **Администраторите** им е възможно е създаване, редактиране и изтриване на продукти.

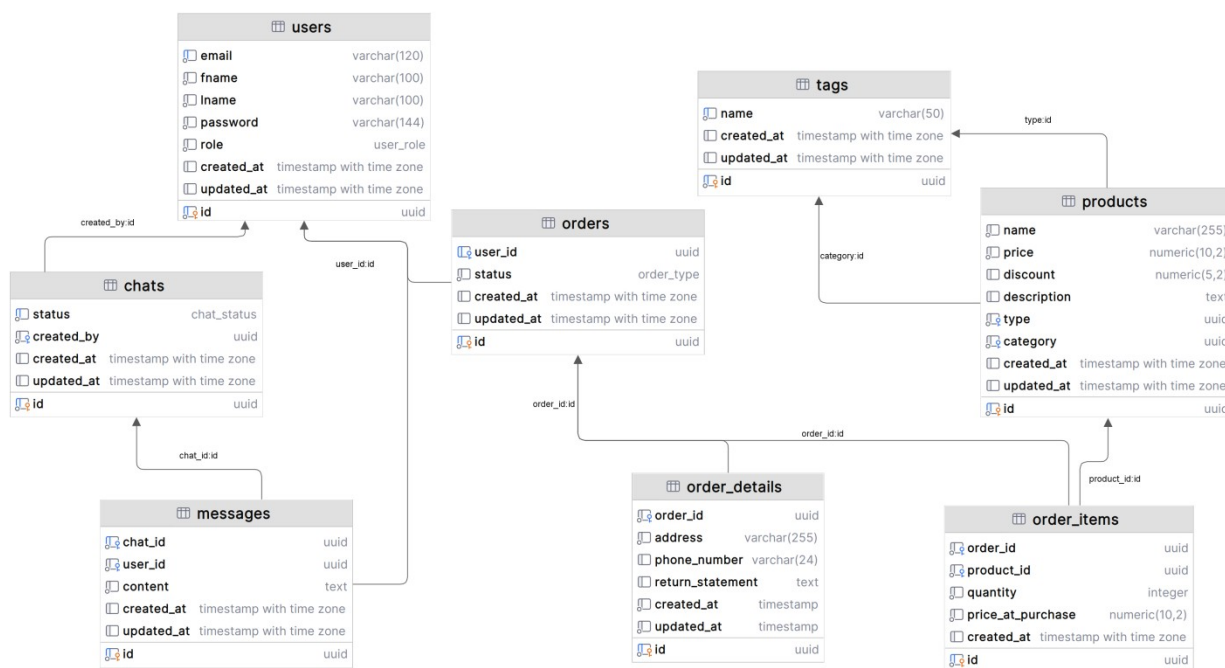
#### **5. Управление на поръчки**

- ✓ **Потребителите** могат да преглеждат своите поръчки и да добавят нови артикули чрез количката.
- ✓ **Администраторите** имат достъп до всички поръчки и могат да ги актуализират или изтриват при необходимост.

#### **6. Връзка с клиентска поддръжка**

- ✓ **Регистрираните потребители** могат да се свържат с екипа по поддръжка в случай на въпроси или проблеми.
- ✓ **Администраторите** могат да отговарят на клиентски запитвания и да предоставят помощ в реално време.

# Модел на данните



Моделът на данните в системата **The Agro Nursery Online Store** е структуриран около основните функционалности на платформата, осигурявайки ефективно управление на потребители, продукти, поръчки и комуникация. Базата данни включва няколко ключови таблици и релации между тях, които гарантират сигурно и оптимизирано съхранение на информация.

## Основни таблици и връзки

### 1. Таблица „users“ (Потребители)

Съдържа информация за регистрираните потребители на платформата, включително техните идентификационни данни, роля и времеви метаданни.

#### ✓ Основни полета:

- id (uuid) – Уникален идентификатор на потребителя
- email (varchar) – Имейл адрес за вход и комуникация
- fname, lname (varchar) – Собствено и фамилно име
- password (varchar) – Хеширана парола
- role (user\_role) – Определя ролята на потребителя в системата (Клиент или Администратор)
- created\_at, updated\_at (timestamp) – Метаданни за създаване и актуализация

## 2. Таблица „products“ (Продукти)

Съхранява информация за наличните продукти в магазина, като включва детайлни характеристики и категоризация.

### ✓ Основни полета:

- `id (uuid)` – Уникален идентификатор на продукта
- `image (varchar)` – Снимка на продукта
- `name (varchar)` – Име на продукта
- `price (numeric)` – Цена
- `discount (numeric)` – Отстъпка (ако има)
- `description (text)` – Подробно описание
- `type (uuid)` – Тип на продукта (реферира към „tags“)
- `category (uuid)` – Категория на продукта (реферира към „tags“)
- `created_at, updated_at (timestamp)` – Метаданни за създаване и актуализация

## 3. Таблица „tags“ (Етикети)

Служи за категоризация и управление на различни типове продукти.

### ✓ Основни полета:

- `id (uuid)` – Уникален идентификатор на етикета
- `name (varchar)` – Име на етикета
- `created_at, updated_at (timestamp)` – Метаданни за създаване и актуализация

## 4. Таблица „orders“ (Поръчки)

Съдържа информация за поръчките на потребителите, като проследява статуси и времеви метаданни.

### ✓ Основни полета:

- `id (uuid)` – Уникален идентификатор на поръчката
- `user_id (uuid)` – Референция към потребителя, който е направил поръчката
- `status (order_type)` – Статус на поръчката (напр. „В обработка“, „Изпратена“)
- `created_at, updated_at (timestamp)` – Метаданни за създаване и актуализация

## 5. Таблица „order\_items“ (Артикули в поръчка)

Свързва поръчките с конкретните продукти, като запазва количеството и цената към момента на покупка.

### ✓ Основни полета:

- `id (uuid)` – Уникален идентификатор
- `order_id (uuid)` – Референция към поръчката
- `product_id (uuid)` – Референция към продукта
- `quantity (integer)` – Закупено количество
- `price_at_purchase (numeric)` – Цена на продукта към момента на поръчката
- `created_at (timestamp)` – Дата на добавяне

## 6. Таблица „order\_details“ (Детайли на поръчка)

Съхранява допълнителна информация за конкретните поръчки, включително адрес и телефонен номер.

### ✓ Основни полета:

- `id (uuid)` – Уникален идентификатор
- `order_id (uuid)` – Референция към поръчката
- `address (varchar)` – Адрес за доставка
- `phone_number (varchar)` – Телефон за контакт
- `return_statement (text)` – Поле за бележки относно връщане
- `created_at, updated_at (timestamp)` – Метаданни за създаване и актуализация

## 7. Таблица „chats“ (Чатове)

Използва се за комуникация между потребителите и администраторите.

### ✓ Основни полета:

- `id (uuid)` – Уникален идентификатор на чата
- `created_by (uuid)` – Потребител, създал чата
- `status (chat_status)` – Статус на чата (Активен, Затворен)
- `created_at, updated_at (timestamp)` – Метаданни за създаване и актуализация

## 8. Таблица „messages“ (Съобщения)

Съхранява съобщенията, изпратени в рамките на чатове.

### ✓ Основни полета:

- `id (uuid)` – Уникален идентификатор
- `chat_id (uuid)` – Референция към чата
- `user_id (uuid)` – Потребител, изпратил съобщението
- `content (text)` – Текст на съобщението
- `created_at, updated_at (timestamp)` – Метаданни за създаване и актуализация

## Връзки между таблиците

1. **users → orders:** Един потребител може да има много поръчки (1:N).
2. **orders → order\_items:** Една поръчка може да съдържа множество продукти (1:N).
3. **order\_items → products:** Един продукт може да бъде в множество поръчки (N:M).
4. **products → tags:** Един продукт може да принадлежи към една категория или тип (N:1).
5. **users → chats:** Един потребител може да има множество чатове (1:N).
6. **chats → messages:** Един чат може да съдържа множество съобщения (1:N).

## Заклучение

Моделът на данните за **The Agro Nursery Online Store** е добре структуриран и обхваща всички ключови аспекти на платформата. Той осигурява:

1. Гъвкаво управление на потребители и роли
2. Проследяване на поръчки и свързаните продукти
3. Категоризация на артикулите с етикети
4. Интерактивна комуникация чрез чат система

## Основни изгледи

Този раздел описва основните изгледи на уеб приложението **The Agro Nursery Online Store**, включително техните функционалности и съответните URI пътища.

### ✓ Начална страница (Home)

- **Описание:** Показва специални оферти и каталога с продукти.
- **URI:** / (пренасочва към /products)

### ✓ Категории продукти (Products Category)

- **Описание:** Показва всички продукти на избраната категория.
- **URI:** /products?type={tag}

### ✓ Детайли за продукт (Product)

- **Описание:** Показва пълна информация за даден продукт и опция за поръчка.
- **URI:** /products/{productId}

### ✓ Обслужване на клиенти (Customer Service - Chat)

- **Описание:** Осигурява интерфейс за чат с поддръжката.
- **URI:** /chats/{chatId} (достъп до конкретен чат)
- **Допълнителен URI:** /chats (списък с всички чатове на потребителя)



### ✓ Регистрация на потребител (User Registration)

- **Описание:** Представя форма за регистрация на анонимни потребители в системата.
- **URI:** /register

### ✓ Вход на потребител (Login)

- **Описание:** Предоставя възможност за влизане в системата.
- **URI:** /login

### ✓ Профил на потребител (User Data)

- **Описание:** Позволява преглед и редактиране на личните данни на потребителя.
- **URI:** /profile (пренасочва към /users/{userId})
- **Допълнителен URI:** /users/{userId} (детайли за конкретен потребител)

## API ресурси (Сървър)

Този раздел описва основните API ресурси на **The Agro Nursery Online Store**. API-ят предоставя CRUD операции за потребители, продукти, поръчки и чат функционалност. Достъпът до някои ресурси е ограничен според ролите на потребителите.

**Забележка:** Тъй като към момента на разработка приложението е изцяло server-side не е необходимо да има разработен API, но все пак в тази секция е предложен един такъв с цел бъдещо развитие на проекта в посока по-развит front-end с възможност за разработка на мобилно приложение.

### 1. Потребители (Users)

- ✓ **Описание:** Получава списък с всички потребители (само за администратори) и позволява създаване на нов потребител.
- ✓ **Методи:**
  - *GET* /api/users – Връща списък с всички потребители.
  - *POST* /api/users – Създава нов потребител (Id се генерира автоматично, връща обновеното entity).
- ✓ **Ограничения:** Само администраторите имат достъп до тази функционалност.

### 2. Потребител (User)

- ✓ **Описание:** CRUD операции за конкретен потребител.
- ✓ **Методи:**
  - *GET* /api/users/{userId} – Връща детайли за конкретен потребител.
  - *PUT* /api/users/{userId} – Обновява информация за потребителя.
  - *DELETE* /api/users/{userId} – Изтрива потребител.
- ✓ **Ограничения:** Само администратори могат да изтриват и редактират личната информация на потребителите.

### 3. Вход (Login)

- ✓ **Описание:** Приема потребителски данни (e-mail и парола) и връща валиден Session Id.
- ✓ **Методи:**
  - *POST* /api/login – Изпраща потребителските данни и връща Session Id.

### 4. Регистрация (Register)

- ✓ **Описание:** Позволява регистрация на нови потребители.
- ✓ **Методи:**
  - *POST* /api/register – Регистрира нов потребител.

### 5. Изход (Logout)

- ✓ **Описание:** Прекратява активната сесия на потребителя.
- ✓ **Методи:**
  - *POST* /api/logout – Излиза от системата и анулира сесията.

### 6. Продукти (Products)

- ✓ **Описание:** Позволява извличане на списък с всички продукти и добавяне на нов продукт.
- ✓ **Методи:**
  - *GET* /api/products – Връща списък с всички продукти.
  - *POST* /api/products – Създава нов продукт (Id се генерира автоматично, връща обновеното entity).
- ✓ **Ограничения:** Само администратори могат да добавят продукти.

### 7. Продукт (Product)

- ✓ **Описание:** CRUD операции за конкретен продукт.
- ✓ **Методи:**
  - *GET* /api/products/{productId} – Връща детайлите за даден продукт.
  - *PUT* /api/products/{productId} – Актуализира продукт.
  - *DELETE* /api/products/{productId} – Изтрива продукт.
- ✓ **Ограничения:** Само администратори могат да редактират и изтриват продукти.

## 8. Чатове (Chats)

- ✓ **Описание:** Позволява получаване на списък с всички чатове и създаване на нов чат.
- ✓ **Методи:**
  - *GET* /api/chats – Връща списък с всички чатове.
  - *POST* /api/chats – Създава нов чат (Id се генерира автоматично, връща обновеното entity).
- ✓ **Ограничения:** Само администратори имат достъп до тази функционалност.

## 9. Чат (Chat)

- ✓ **Описание:** CRUD операции за конкретен чат.
- ✓ **Методи:**
  - *GET* /api/chats/{chatId} – Връща детайли за конкретен чат.
  - *PUT* /api/chats/{chatId} – Обновява информация за чата.
  - *DELETE* /api/chats/{chatId} – Изтрива чат.

## 10. Съобщения (Messages)

- ✓ **Описание:** Позволява получаване на всички съобщения в даден чат и изпращане на ново съобщение.
- ✓ **Методи:**
  - *GET* /api/chats/{chatId}/messages – Връща всички съобщения от даден чат.
  - *POST* /api/chats/{chatId}/messages – Изпраща ново съобщение (Id се генерира автоматично).
- ✓ **Ограничения:** Само администратори имат достъп.

## 11. Съобщение (Message)

✓ **Описание:** CRUD операции за конкретно съобщение.

✓ **Методи:**

- `GET /api/chats/{chatId}/messages/{messageId}` – Връща детайли за конкретно съобщение.
- `PUT /api/chats/{chatId}/messages/{messageId}` – Обновява съобщение.
- `DELETE /api/chats/{chatId}/messages/{messageId}` – Изтрива съобщение.

## 12. Поръчки (Orders)

✓ **Описание:** Позволява получаване на всички поръчки и създаване на нова поръчка.

✓ **Методи:**

- `GET /api/orders` – Връща списък с всички поръчки.
- `POST /api/orders` – Създава нова поръчка (Id се генерира автоматично, връща обновеното entity).

✓ **Ограничения:** Само администратори имат достъп до тази функционалност.

## 13. Поръчка (Order)

✓ **Описание:** CRUD операции за конкретна поръчка.

✓ **Методи:**

- `GET /api/orders/{orderId}` – Връща детайли за дадена поръчка.
- `PUT /api/orders/{orderId}` – Актуализира информация за поръчка.
- `DELETE /api/orders/{orderId}` – Изтрива поръчка.

✓ **Ограничения:** Само администратори имат достъп до тази функционалност.

# Използвани технологии

### 1. User-Friendly Interface

За създаване на **интуитивен и лесен за навигация интерфейс** са използвани **templ, tailwindcss**. Тези технологии осигуряват динамично и ефективно изграждане на дизайн елементи.

### 2. Real-Time Customer Support

Внедрена е система за **поддръжка на чат в реално време**, използваща **websockets**. Технологията позволява на потребителите да се свързват

незабавно с представители на администрацията, за да получават бързо съдействие по всяко време.

### 3. Secure User Registration

За сигурността на потребителските данни е използвана система за **регистрация и управление на профилите**, базирана на **gorilla/sessions** като удостоверителна услуга. Потребителите могат да се регистрират, управляват личните си данни и следят своите поръчки в защитена среда.

### 4. Dynamic Routing

За **осигуряване на гладка и безпроблемна навигация** между различните страници на сайта се използва **gin**, който отговаря за сървърната маршрутизация. Технологиата гарантира ефективна обработка на заявките и бързо зареждане на страниците.

### 6. Robust Database Management

За ефективно управление на базата данни е използвана комбинация от **sqlc** и **pgx** за взаимодействие с **PostgreSQL базата данни**. **sqlc** генерира типизирани SQL заявки директно от Go код, което осигурява по-висока производителност и безопасност при взаимодействието с базата данни. От своя страна, **pgx** предоставя бърз и ефективен интерфейс за работа с PostgreSQL, като поддържа асинхронни операции и опции за оптимизация на производителността.

## Архитектура на приложението

Архитектурата на проекта е изградена върху принципите на **Three-Tier Architecture**, като всяка основна част от приложението е разделена на три слоя, за да се осигури модулност, мащабируемост и лесна поддръжка.

#### 1. Frontend (Потребителски интерфейс):

1.1. Използва се **templ**, **tailwindcss** и за изграждане на динамичен и отзивчив интерфейс, който позволява лесно навигиране през продуктите и бърза обработка на действията от потребителя.

#### 2. Backend (Сървърна част):

2.1. Използва се **Gin** като основен web router за маршрутизация, осигурявайки ефективност при обработка на HTTP заявки и бързо зареждане на страниците.

2.2. Основният контролер и логика на бизнес процесите се намират тук, включително обработката на поръчки, потребителски данни и взаимодействия с базата данни.

2.3. **WebSockets** се използва за осигуряване на чат функционалност и реализиране на чат поддръжка в реално време.

#### 3. Database (База данни):

3.1. Използва се **PostgreSQL** като основна релационна база данни за съхранение на данни за потребители, продукти, поръчки и чатове.

3.2. Взаимодействието с базата данни е осигурено чрез **sqlc** и **pgx**, които гарантират ефективност и сигурност при изпълнение на SQL заявки, като същевременно осигуряват лесна и бърза работа с данни.

#### 4. Сигурност и удостоверяване:

4.1. За управление на потребителските сесии и сигурността на платформата се използва **pgstore** базиране на **gorilla/sessions** с **gorilla/securecookie**, което осигурява стабилно и безопасно удостоверяване на потребителите.

4.2. Регистрацията и управлението на данни за потребители се реализират с контрол върху правата и ролите (Клиент и Администратор).

#### 5. API (Backend API):

5.1. API-то на сървъра предлага различни RESTful ресурси, които позволяват взаимодействие със сървъра за управление на потребители, продукти, поръчки и чатове.

5.2. Основните операции за работа с данни включват **GET**, **POST**, **PUT** и **DELETE** методи, както и ролеви ограничения (само за администратори).

## Конфигурация на приложението

Конфигурацията на приложението обхваща всички аспекти, свързани с настройките на сървърната част, база данни, сигурност и различни допълнителни опции за персонализиране на поведението на системата.

#### 1. Сървърна конфигурация:

1.1. Използваният **Gin** framework осигурява бърза и ефективна маршрутизация на HTTP заявки. Конфигурацията на Gin се извършва чрез настройка на основните пътища (routes), middleware, както и обработка на грешки и логове.

1.2. Използване на **WebSockets** за чат комуникацията, като конфигурацията позволява бърза връзка и поддръжка на множество активни сесии за чат в реално време.

#### 2. База данни (PostgreSQL):

2.1. Базата данни се настройва чрез **sqlc** за генериране на типизирани SQL заявки и с помощта на **pgx** за оптимизирано взаимодействие с PostgreSQL. Конфигурацията включва настройка на връзките към базата данни, както и управлението на миграции и транзакции.

2.2. Конфигурацията на PostgreSQL включва настройки за връзка, време на изчакване на заявките, както и максимален брой връзки за оптимална производителност.

#### 3. Сигурност:

- 3.1. Конфигурацията на **pgstore** осигурява сигурно управление на сесиите, като позволява съхранение на сесиите в сигурни cookie файлове и в базата данни.
- 3.2. За конфигуриране на защитен достъп и удостоверяване се използват подходящи алгоритми за хеширане на пароли и сесии.

## Източници на информация

*За разработката на приложението и текущата документация като външни ресурси са използвани документациите на всяка една от гореописаните библиотеки.*