**Online Shop for Computers and Computer Components (CCC)**

**Документация**

**Автор: Димитър Манчев, 62355**

Contents

[1. Въведение 3](#_Toc107716952)

[2. Използвани технологии. 3](#_Toc107716953)

[2.1. FrontEnd 3](#_Toc107716954)

[2.2. Backend 3](#_Toc107716955)

[3. Роли 3](#_Toc107716956)

[3.1. Гост (нерегистриран потребител) 3](#_Toc107716957)

[3.2. Клиент (регистриран потребител) 4](#_Toc107716958)

[3.3. Персонал (staff) 4](#_Toc107716959)

[3.4. Администратор 4](#_Toc107716960)

[4. Функционални изисквания. 4](#_Toc107716961)

[5. База данни 5](#_Toc107716962)

[6. Структура 5](#_Toc107716963)

[6.1. Frontend 5](#_Toc107716964)

[6.2. Backend 6](#_Toc107716965)

# Въведение

В днешно време пазаруването в интернет се превръща в доста важна част от живота на всеки живот. Необходимостта от добри системи, които да предоставят удобно и лесно онлайн пазаруване предизвиква голям интерес в създаването на най-добрият възможен уеб сайт. Затова CCC е система, която ще осигури удобно, бързо и надеждно онлайн пазаруване на компютри и компютърни части.

# Използвани технологии.

## FrontEnd

* React.js
* Bootstrap
* Formik
* Yup

## Backend

* Node.js
* Express.js
* Pg-promise – библиотека за изпращане на заявки към PostgreSQL база от данни, базирана на “обещания” (promises)
* Nodemailer – node модул, служещ за изпращане на имейли, използвайки SMTP протокол и google services
* Multer – middleware за обработване на получени файлове (png, xlsx, csv и др.)

# Роли

## Гост (нерегистриран потребител)

Гостът на системата може да преглежда всички налични продукти в магазина и да ги добавя към своя “Умна Количка“. Целта на умната количка е да съхранява всички продукти, които гостът е отбелязал. При опит за създаване на поръчка, гостът е пренасочен към страница за влизане/регистриране в системата.

Също така гостът може да попълни формуляр за връзка с персонала на системата, при наличие на въпрос или запитване.

## Клиент (регистриран потребител)

Клиентът може да създава и плаща поръчки, да преглежда своят профил, да редактира личните си данни, както и да преглежда историята на поръчките си. Притежава и всички права, които има гостът.

Преди да бъде напълно активиран клиентския акаунт, той трябва да бъде верифициран, като за целта се изпраща имейл до посочената електронна поща, съдържаща линк. При натискане на линка, потребителският акаунт се верифицира и получава достъп до системата.

## Персонал (staff)

Персоналът има достъп до “back-office” частта на системата. Всеки член на персонала на магазина има зададени права, които му дават достъп до различни интерфейси на системата – например интерфейс за продукти, поръчки, справки и т.н. В зависимост от получените права, той може да добавя/променя/деактивира продукти, да променя статус на поръчки, да генерира справки, да отговаря на тикети, да добавя нови staff акаунти и др.

Важно е да се отбележи, че персоналът на системата **НЯМА** достъп до “front-office” на системата, т.е. страницата с продукти, количката, историята на поръчките и страницата с личния профил. Също така не съществува регистриране на staff акаунт. Такъв тип акаунт може да се създаде само от администратор или член на персонала, който притежава съответните права.

## Администратор

Администраторът има достъп до всички интерфейси на системата и има всички права. Той може да създава нови роли, както и да задава/премахва роли на персонала. Администраторът може да блокира клиенти от магазина, при наличие на добро основание за това.

# Функционални изисквания.

* Регистрация/вход в системата. (гост)
* Преглеждане на наличните продукти в магазина. (гост, клиент)
* Добавяне/премахване на продукт към количката. (гост, клиент)
* Промяна на количеството на продукти в количката. (гост, клиент)
* Достъпване на историята на поръчките. (клиент)
* Достъпване на личния профил. (клиент)
* Създаване на поръчка. (клиент)
* Плащане на поръчка. (клиент)
* Добавяне/премахване/модифициране на продукти в системата. (администратор, персонал)
* Промяна на статус на поръчка. (администратор, персонал)
* Създаване/премахване/редактиране на staff акаунт. (администратор)
* Създаване на роли и задаване на права. (администратор)
* Блокиране на клиент. (администратор)

# База данни

Използваната база от данни е PostgreSQL v14.1.

Схема на базата от данни:

<https://drive.google.com/file/d/1xJ63p_h1zSQBYwxrrtQX00TJOT21reRG/view?usp=sharing>

# Структура

## Frontend

* App.js – главен файл/свързващ компонент, съдържащ React Router и проверяващ потребителските сесии.
* Front-office
* Nav.js – навигационна лента
* Product.js – страницата с продукти
* Shopcart.js – количка
* Checkout.js – страница за създаване на поръчка
* Account.js – страница за личен потребителски профил
* Paypal.js – страница пренасочваща към PayPal gateway
* OrdersHistory.js – страница, съдържаща историята на поръчки на потребителя
* DetailedOrder.js – страница, визуализираща повече информация за дадена направена поръчка.
* Voucher.js – съдържа ваучер компонента (**BUG! NOT WORKING!)**
* NotVerified.js – страница, визуализираща се при неверифициран потребител
* Login.js – страница за вход в системата (клиент)
* Register.js – страница за регистриране (клиент)
* Forbidden.js – страница, визуализираща се при достъпване на забранен ресурс
* NotFound.js – страница, визуализираща се при достъпване на несъществуващ ресурс
* Back-office
* BackOfficeLogin.js – страница за вход в системата (персонал, администратор)
* BackOfficeNav.js – навигационна лента
* ProductManagement.js – CRUD интерфейс за продукти
* OrderManagement.js – интерфейс показващ всички поръчки
* UsersManagement.js – интерфейс, показващ всички клиенти
* StaffManagement.js – CRUD интерфейс за акаунти на персонала
* TargetGroups.js – интерфейс, показващ всички целеви групи
* TargetGroupCreate.js – страница за създаване на целева група
* Promotions.js – интерфейс, показващ всички налични промоции
* PromotionCreate.js – страница за създаване на промоция
* EmailTemplate.js – страница за конфигуриране на изглед на имейл, изпратен при успешно създадена/платена поръчка.
* Roles.js – интерфейс, показващ всички налични роли

## Backend

* App.js – основен файл, свързващ всички middleware, стратиращ сървъра.
* db.js/\_db.js – файлове, съдържащи връзката към PostgreSQL базата от данни.
* dictionary.js – файл, съдържащ основни думи за превод в системата.
* utils.js – файл, съдържащ всички основни функции, които се използват от всички останали файлове
* paypalClient.js, paypalCreateOrder.js, paypalTransaction.js – файлове, обработващи постъпила PayPal трансакция
* \*Router.js – всички router файлове
* config.json – конфигурационен файл за имейли
* global.conf.js – основния конфигурационен файл, съдържащ всички конфигурации за магазина
* URLDispatchTable.js – основен файл, грижещ се за предотвратяването на достъпването на забранен ресурс за потребителя, който извиква заявката.
* Папка **cert -** файловеза SSL сертификат, позволяващ ни локална HTTPS връзка.
* Папка **images** - всички налични снимки в магазина.
* Папка **customErrors** -персонализирани системни грешки, помагащи за обработка на грешки и връщане на подходящо съобщение на потребителя.
* Папка **fonts** – допълнителни шрифтове