**Денис Велиев**

**F104630**

**CSCB532 Практика по програмиране и интернет технологии**

**Приложение „Логистична компания“**

**Задание**

Да се реализира уеб приложение „Логистична компания“, което да служи за управление на процесите в логистична компания. Основната дейност на компанията е да осъществява услуги по приемане и доставяне на пратки. Компанията разполага с офиси на различни места и в нея работят два вида служители: куриери (извършват доставката на пратките) и офис-служители (обслужват клиенти в офисите на компанията). Клиентите на компанията изпращат и/или получават пратки от офисите на компанията или от друг адрес. Пратките имат подател, получател, адрес за доставка и тегло. За определяне на цената на за доставка играе роля теглото на пратката и дали тя ще се доставя до офис или до точен адрес. Доставките до офис са по-евтини, отколкото тези до адрес. Служителите на компанията могат да виждат всички пратки, които са регистрирани в системата. Всеки клиент може да вижда само пратките, които е изпратил, получили или очаква да получи.

**Функционални изисквания на системата**

Уеб приложението „Логистична компания“ трябва да включва минимум следните функционалности:

1. Регистриране на потребители и вход в системата

2. Възможност за задаване на роли на потребителите (служител, клиент)

3. Въвеждане, показване, редактиране и изтриване на данни за:

a. Логистична компания

b. Служител на компания

c. Клиент на компания

d. Офис на компания

e. Пратка

4. Служителите на компанията трябва да могат да регистрират изпратените и получените

пратки.

5. Справки за:

a. Всички служители в компанията

b. Всички клиенти на компанията

c. Всички пратки, които са били регистрирани

d. Всички пратки, които са регистрирани от даден служител

e. Всички пратки, които са изпратени, но не да получени

f. Всички пратки, които са изпратени от даден клиент

g. Всички пратки, които са получени от даден клиент

h. Всички приходи на фирмата за определен период от време

6. Всеки служител може да вижда всички пратки

7. Всеки клиент може да вижда пратките, които е изпратил или получил

**Технологични изисквания**

Приложението трябва да бъде уеб базирано, с възможност за визуализиране в найпопулярните браузери (Chrome, Mozilla Firefox, Internet Explorer). Дизайнът му трябва да бъде респонсив (подходящ за визуализиране и на мобилни устройства). Разработената система трябва да се състои от код, база данни и документация. Кодът трябва да включва подробни коментари. Документацията трябва се състои от детайлно описание на функционалностите на системата, включително екрани, на които ясно се вижда коя част от програмата за какво се използва.

**Документация**

1. **Въведение**

Уеб приложението „Логистична компания“ е разработено с цел управление на процесите в логистична компания, специализирана в приемането и доставянето на пратки. Системата обединява служители и клиенти на компанията, като предоставя интуитивен интерфейс за регистрация на пратки, управление на клиентски и служителски профили, както и извършване на различни справки относно статуса на пратките.

1. **Използвани технологии**

Уеб приложението е разработвано чрез използване на различни технологии.

За “front-end” частта са използвани:

* **React с Vite**
* **Material UI** - библиотека за готови компоненти
* **ReduxJS Toolkit** - за глобален “state” мениджмънт
* **Axios** - за правене на заявки към “back-end” частта
* **React Router Dom** - за навигиране между страниците
* **TailwindCSS** - за по-лесна стилизация на някои компоненти

За “back-end” частта са използвани:

* **ExpressJS** -уеб фреймуърк за Node.js, който предоставя мощен набор от функции за уеб и мобилни приложения.
* **bcryptjs** - за хеширане на пароли преди тяхното съхранение в базата данни
* **cors** - за конфигуриране на Cross-Origin Resource Sharing (CORS), позволява на уеб приложенията да правят заявки към сървъра от различен домейн.
* **dotenv** - за зареждане на конфигурационни променливи от .env файл в Node.js проекта, което помага за сигурното съхранение на чувствителни данни.
* **express-validator** - за валидация и санитизация на входящите данни, използва се с ExpressJS за увеличаване на сигурността и надеждността на приложението.
* **jsonwebtoken** - за създаване и валидация на JSON Web Tokens, използва се за аутентикация и управление на сесии в уеб приложения.
* **mongoose** - ODM (Object Data Modeling) библиотека за MongoDB, която улеснява работата с бази данни и предоставя схеми за структуриране на данните.
* **uuid** - за генериране на уникални идентификатори, полезно за създаване на уникални ключове в бази данни или за идентификация на транзакции и сесии.