***Кратка документация***

***CR-Logistics***

# Задание

*Да се направи уеб приложение с база данни, в което могат да се добавят и редактират:*

* *Населени места*
* *Пътища (Начален град, Краен град)*
* *Логистичен център*

*През интерфейса на приложението трябва да могат да се добавят и редактират населените места и пътища.*

*При натискане на бутон да се изпълни следния алгоритъм:*

*Даден е район с N населени места (броя на въведените градове в базата) и M пътища (броя на пътищата в базата), всеки откоито свързва две различни населени места на района.*

*Да се избира такова населено място C, че най-отдалеченото от останалите населени места да е колкото може по-близо до C.След като се избере такова място C да се запише в базата в таблицата като логистичен център. Трябва да има валидация, акопри натискане на бутона няма промяна в населените места и пътищата (т.е. били са същите като последното изпълнение), да не бъде създаван нов логистичен център.*

*Допълнително условие (незадължително): След изпълнението на алгоритъма да се визуализира с Highcharts или друг framework по избор населените места, пътищата и логистичните центрове.*

# Компоненти на направеното решение

Текущото уеб приложение има следната структура

* База данни – създадена по Code first концепция на Entity Framework
* Back-end – създаден на следните нива:
  + DataModel – дефинира основните модели данни и съответните им интерфейси;
  + DataAccess – осъществява връзка с база данни и изпълнява задачи, свързани с добавяне и обновяване (както е упоменато в заданието) на записи в нея;
  + DataService – предоставя интерфейси свързани с управлението на различни обекти – в случая CityService & RouteService;
  + WebApp – предоставя приложен интерфейс за комуникация.
* Front-end – приложение, дефинирано с Angular 9, което предоставя следните компоненти:
  + Components part – дефинира logistic-center като основен компонент, в който са включени компонентите за управление на градове и пътища;
  + Services – дефинира injectable обекти, чрез които могат да се осъществяват Http заявки към API в back-end частта;
  + Models – дефинира обектите, с които приложението работи.

# Използвани алгоритми

* Алгоритъм на Прим за намиране на минимално покриващо дърво
* Алгоритъм за намиране на коефициент за централност

# Възможности за обновяване

* Валидационни съобщения – в процес на доработване;
* Още алгоритми – класът Algorithms остава отворен за имплементирането на още алгоритми за намиране на Логистичен център;
* Визуализация – подобряване на изгледите;