



به نام خدا

پروژه درس ساختمان داده  
استاد درس : دکتر رحمانی  
تی ای پروژه : خانم زهرا امیری  
نام افراد : نرگس مشایخی ، ماهان محمودی کیسمی  
شماره دانشجویی افراد : ۹۸۴۱۲۴۹۹-۹۸۵۴۲۴۳۶  
ترم ارایه درس : نیمسال اول سال ۱۴۰۰-۱۴۰۱

## چکیده :

با توجه به نیاز پزشکان و روند پیشرفت علم به سمت کامپیوتری شدن هرچه بیشتر کارها به جای انجام دستی آنها ، این پروژه تعریف شده است . به این شکل که پزشک میتواند با سرعت بالا و حجم فضای اشغال شده بهینه از کامپیوتر ، برای هر بیماری داخل دیتا ست دارو تجویز کند و اطلاعات مربوط به تداخلات دارو های داخل نسخه باهم و تداخلات دارویی با بیماری بیمار را مشاهده کند تا بتواند بهترین تصمیم را بگیرد .

## اهداف و راهکار ها :

هدف اصلی این پروژه انتخاب و یا طراحی ساختمان داده ای برای مسئله طرح شده بالا است که بتواند دو شرط اصلی زیر را پیاده سازی کند :

- توابع با سرعت بالا پاسخ به نیاز های مطرح شده بدهند
- حافظه اشغال شده برنامه در بهینه ترین حالت باشد

در حقیقت هدف اصلی سوال طراحی ساختمان داده ای بود که بتواند توازن را بین دو شرط مطرح شده برقرار کند و در کنار سرعت بالای برنامه حافظه کمی اشغال کند .

### راهکار :

برای حل این مسئله نیاز هست که  $time\ complexity$  ,  $space\ complexity$  ساختمان داده های مختلف تحت بررسی قرار بگیرد که بتوان بهترین ساختمان داده را انتخاب نمود یا با ترکیب ویژگی های ساختمان داده های مختلف ساختمان داده جدیدی طراحی کرد که هدف مسئله را به بهترین وجه پیاده سازی کند .

در این مسئله با توجه به توابعی که نیاز به تعریف آنها بود در اکثر بخش ها نیاز به سرچ کردن یک داده بین داده های داخل دیتا بیس وجود دارد بنابر این در انتخاب ساختمان داده باید به این نکته توجه داشت که سرچ کردن دیتا در هر ساختمان داده از چه اوردر زمانی ای پیروی میکند .

همین طور نیاز به ذخیره کردن داده ها به شکل تاپل یا دوتایی های مرتب بود .

با توجه به نیاز های مطرح شده مسئله لیست و آرایه از بین گزینه های کاندید برای ساختمان داده اصلی مسئله حذف شدن ، در ادامه با توجه به مقایسه ساختمان داده ها به دو ساختمان داده دیکشنری و هشتتیل رسیدیم که هم امکان ذخیره به صورت دوتایی را دارند هم سرعت بالایی در سرچ کردن داده داشتن .

در مقایسه این دو ساختمان داده ساختمان داده هشت تیل از سرعت بیشتری در سرچ برخوردار بود اما اوردر زمانی هردو ساختمان داده مشابه بود و به همین دلیل با توجه به مسایلی نظیر مشکلات در کالیزن و هش کردن دیتا و دیکد کردن دیتا در زمان ذخیره سازی ، هشت تیل هم از گزینه ها کنار رفت و ساختمان داده دیکشنری به عنوان ساختمان داده اصلی پروژه انتخاب شد ، البته در بخش های مختلف با توجه به کارکرد توابع و ویژگی های دیتا های ذخیره ای از ساختمان داده های دیگر نظیر لیست هم استفاده شده است .

دومین مورد پر اهمیتی که در این پروژه وجود داشت نحوه ذخیره سازی اطلاعات در رم و برنامه بود .

### راهکار:

برای حل این مسئله ، دو راهکار اصلی وجود دارد :

- ذخیره سازی داده ها به صورت خردخرد در حین برنامه ، مثلاً بعد از هر تغییر زمانی که دیتا در دیتاست برنامه تغییری میکند همان تغییر عیناً در فایل ها هم اعمال شود.
- ذخیره سازی اطلاعات به صورت کلی در پایان پروژه .

هر کدام از راهکار های بالا دارای مزایا و معایبی هستند که باید با توجه به آنها بین دو روش یکی را انتخاب نمود . روش اول مزیتی که نسب به روش دوم دارد این است که نیاز نیست دیتا هایی که در برنامه تغییری نکردن دوباره نوشته شوند در فایل و تنها دیتا های تغییر کرده را میتوان در فایل تغییر داد ، اما این روش معایبی هم به همراه دارد که از جمله آنها میتوان به افزایش زمان توابع اشاره کرد .

روش دوم مزیتی که نسبت به روش اول دارد این است که تمام تغییرات یک بار و در انتهای برنامه اعمال میشود از همین جهت سرعت توابع بیشتر است نسبت به روش اول . با توجه به انتخاب ساختمان داده دیکشنری در برنامه و دانستن این نکته که دیکشنری ترتیب دیتا را حفظ نمیکند ، دیتا بیس برنامه ترتیب متفاوتی از دیتابیس داخل فایل ها داشت و این نکته پیدا کردن دیتا خاص را با آوردن یک غیر ممکن میکرد و همین باعث میشد که برای هر نیاز باشد بخش زیادی از دیتا های داخل فایل ها را جستجو کرد . به دلایل گفته شده در بالا روش دوم روش بهینه تری بود و در پروژه از این روش برای ذخیره سازی دیتا استفاده شده است .

### **داده ها و نتایج :**

این برنامه از ۴ دیتاست اصلی استفاده میکند :

- Drugs
- Diseases
- Affects
- Allergies

که این داده ها به صورت فایل در اختیار برنامه قرار میگیرد ، برنامه در ابتدا آنها را در حافظه اش ذخیره میکند و تغییرات را بر روی حافظه خود اعمال میکند و در هر مرحله با توجه به نیاز کاربر ریسپانس مناسب را از همین دیتا ست داخل برنامه به کاربر نمایش میدهد .

در آخر برنامه تمام اطلاعات نوشته شده در حافظه برنامه را مجدد در فایل ها مینویسد تا برای استفاده مجدد از برنامه در دفعات بعد فایل ها آپدیت شده باشند.