به نام خدا

برنامه سازی پیشرفته

(گزارش پروژه)

دانشکده ریاضی و علوم کامپیوتر

نیمسال دوم ۰۴-۰۳

CodeScraper

استاد درس : دکتر کارلو آبنوسیان

اعضای گروه: هومان حسین‌پور – ایلیا حبیبی – محمد احمدی – سروش میرشکاری

فعالیت‌های اعضای تیم:

محمد احمدی مسئولیت بخش Detector را بر عهده داشت که شامل استخراج توکن‌های مربوط به آگهی‌ها از دو سایت مسکن و ملکمون می‌شد. برای سایت مسکن از کتابخانه Selenium جهت استخراج HTML استفاده شد، چرا که این سایت فاقد API بود. در مورد سایت ملکمون، با وجود استفاده از API سایت از طریق پکیج requests، به دلیل محدودیت‌های دسترسی مستقیم با توکن‌ها، راهکار جایگزینی برای استخراج کامل اطلاعات آگهی‌ها پیاده‌سازی شد.

ایلیا حبیبی بخش‌های Scraper و الگوریتم مشابهت را توسعه داد. در بخش Scraper، مشکل دریافت ناقص داده‌ها از ملکمون با تغییر استراتژی به دریافت مستقیم JSON از صفحه اصلی حل شد. برای الگوریتم مشابهت، از SequenceMatcher کتابخانه difflib استفاده گردید که به دلیل سادگی پیاده‌سازی و دقت مناسب در مقایسه رشته‌های متنی انتخاب شد. چالش‌های فنی شامل مشکلات اتصال به پایگاه داده بود که با انجام اصلاحات لازم برطرف گردید.

هومان حسین‌پور طراحی و پیاده‌سازی پایگاه داده را انجام داد که شامل دو جدول اصلی برای اطلاعات آگهی‌ها و روابط مشابهت بین آنها بود. برای بهبود ساختار، پیشنهاد ایجاد جدول واسط جهت برقراری ارتباط چندبهچند بین جداول ارائه شد که می‌توانست بازیابی اطلاعات را تسهیل نماید.

سروش میرشکاری مسئولیت بخش Cleaner را عهده‌دار بود که شامل پیاده‌سازی قواعد regex برای یکسان‌سازی و استانداردسازی داده‌های ورودی می‌شد. تمرکز اصلی بر کاهش حجم کد و افزایش خوانایی آن بود که با موفقیت محقق گردید.

مشکلات و چالش‌های پروژه:

در روند توسعه پروژه با چالش‌های متعددی مواجه شدیم. تغییرات مکرر ساختار صفحات وبسایت‌ها، محدودیت‌های نرخ درخواست از سمت سرورها، و تفاوت‌های ساختاری در داده‌های دو منبع اصلی از جمله این مشکلات بودند. همچنین،

چالش‌های فنی در اتصال به پایگاه داده و اجرای الگوریتم‌ها وجود داشت که با همکاری تیمی و تست‌های مکرر برطرف گردید.

تحلیل معماری سیستم:

معماری سیستم مبتنی بر رویکرد شیءگرا طراحی شده که قابلیت توسعه و تغییرات آینده را فراهم می‌سازد. با این حال، در بخش پایگاه داده می‌توان با ایجاد جدول واسط، ساختار بهتری برای ارتباط بین موجودیت‌ها ایجاد کرد. الگوریتم مشابهت نیز در صورت افزایش حجم داده‌ها نیاز به بهینه‌سازی خواهد داشت.

قابلیت توسعه و گسترش:

سیستم حاضر از قابلیت توسعه پذیری مناسبی برخوردار است. امکان افزودن منابع داده جدید، توسعه الگوریتم‌های پردازشی پیشرفته‌تر، و ایجاد واسط‌های کاربری تکمیلی وجود دارد. همچنین، معماری ماژولار سیستم این امکان را فراهم می‌سازد که هر بخش به صورت مستقل ارتقا یابد.

چالش‌های استفاده واقعی:

در صورت بکارگیری این سیستم در محیط عملیاتی، چالش‌هایی مانند وابستگی به ساختار صفحات ثالث، نیاز به بروزرسانی مداوم الگوریتم‌ها، و ملاحظات حقوقی در استفاده از داده‌های وبسایت‌ها مطرح خواهد شد. همچنین، مقیاس‌پذیری سیستم در مواجهه با حجم بالای داده نیاز به توجه ویژه دارد.

توجیه انتخاب‌های فنی:

انتخاب Selenium برای صفحات دینامیک، استفاده از API در موارد ممکن، و بکارگیری SequenceMatcher برای مقایسه رشته‌ها، همگی بر اساس ارزیابی دقیق نیازهای پروژه و مزایای هر فناوری صورت پذیرفت. معماری شیءگرا نیز به دلیل مزایای آشکار در توسعه‌پذیری و نگهداری انتخاب شد.