实验报告

刘雅迪
计 26

一、爬虫

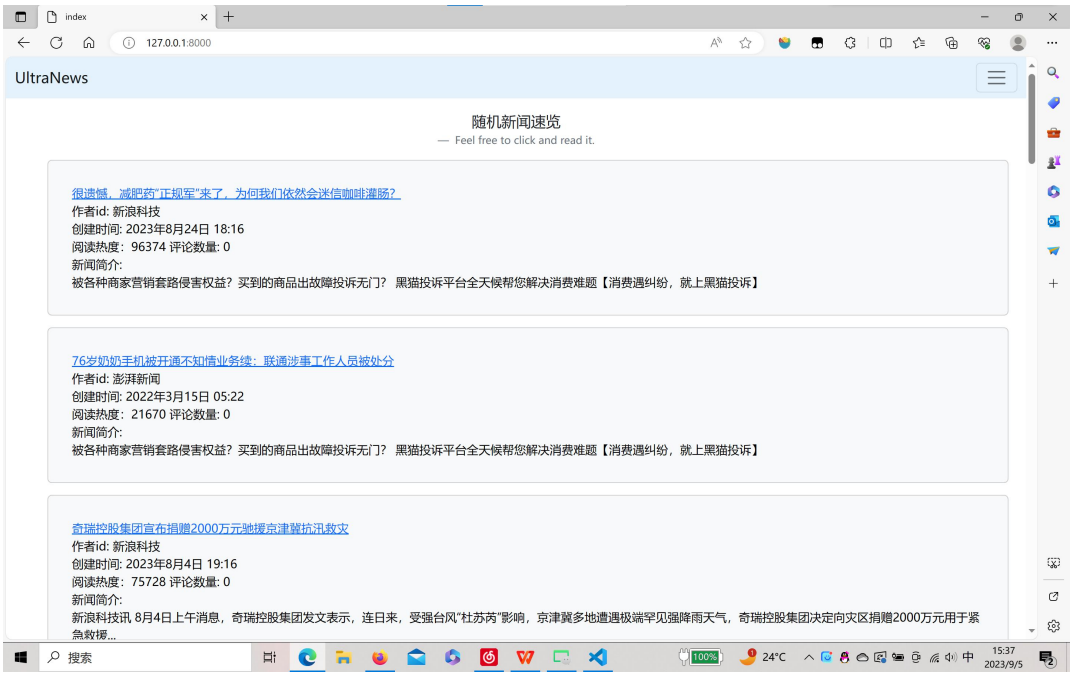
从新浪新闻的科技板块中选取，通过寻找网站的 API 爬取数据。

共爬取 5162 条新闻数据。爬取信息包括新闻的标题、作者、创建时间、新闻简介、正文文本、网页 URL 和新闻中含有的图片（具体代码见 crawler.py），爬取的数据被保存在 result.txt 文件中。此外，新闻中含有的图片被下载到本地保存（具体代码见 load_imgs.py）

二、Web 系统设计

使用基于 python 语言的 Django 框架实现了新闻网站的搭建。网站要求的首页、新闻列表页、新闻正文页、分类页和分类列表页、搜索结果页均已满足，并且利用 bootstrap 的组件实现了网站的简单美化。

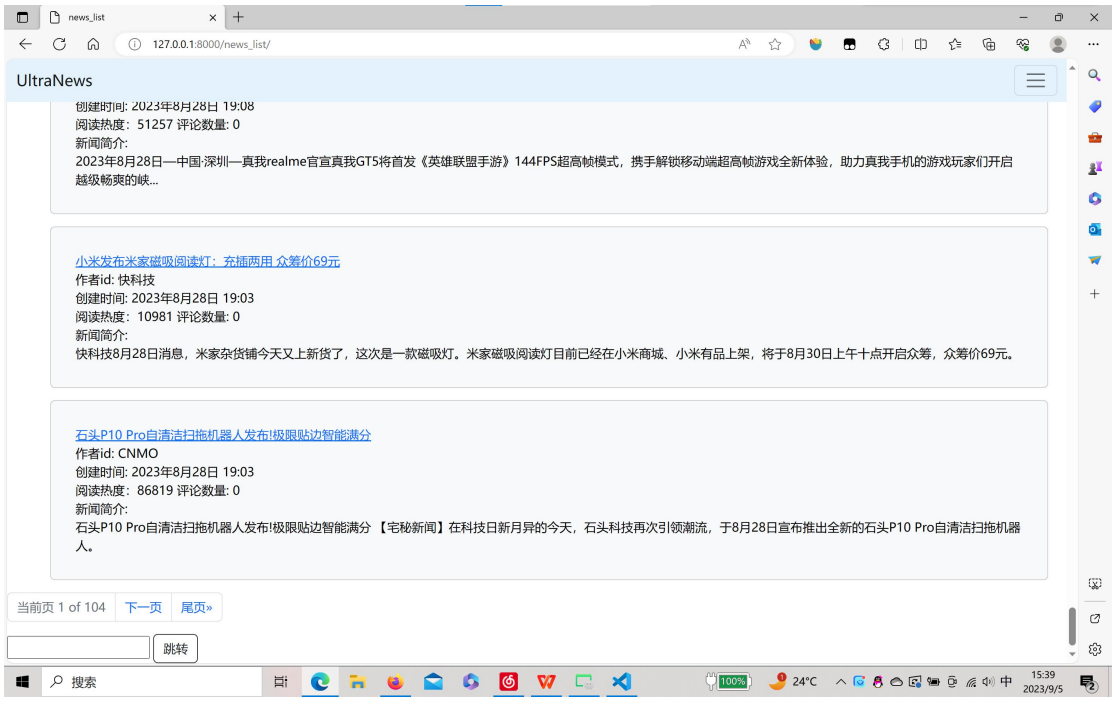
首页实现了随机新闻展示、搜索以及跳转到其他页面的功能。添加了“更新”按钮，点击则会更新首页的随机新闻。此外为了首页的美观，使用 offcanvas navbar 将搜索框放置在了画布外。



首页随机新闻展示

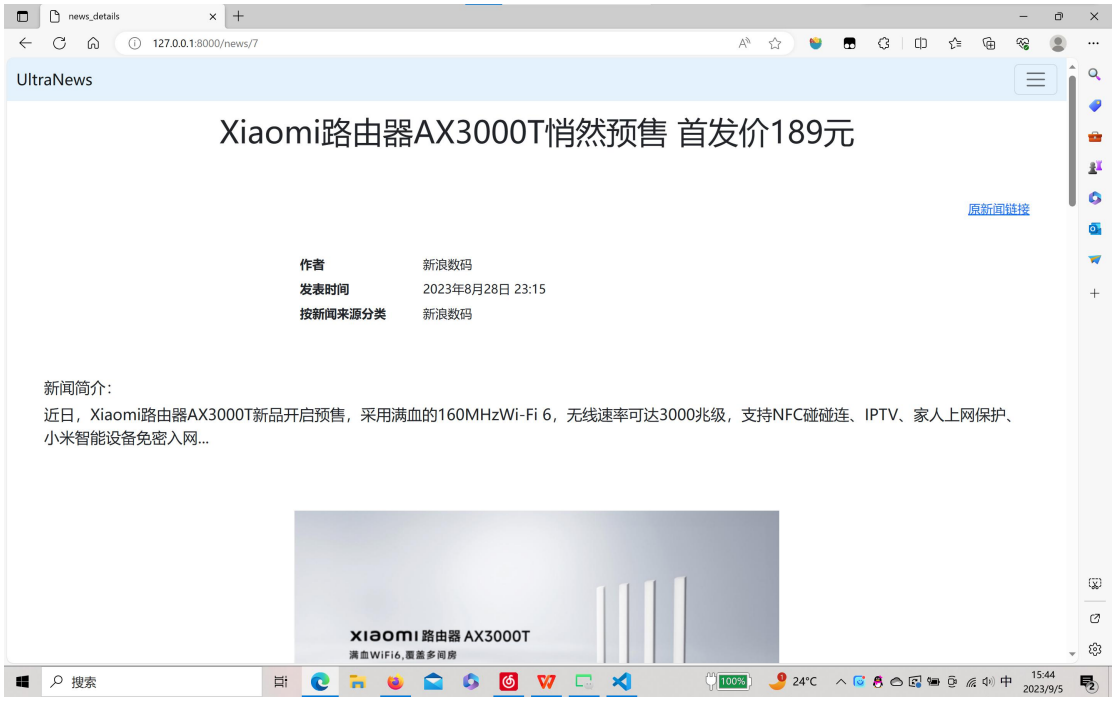
首页搜索框展示

新闻列表页通过分页方式列出了系统中的所有新闻，且点击列表页中的条目可以跳转到对应的新闻正文页。

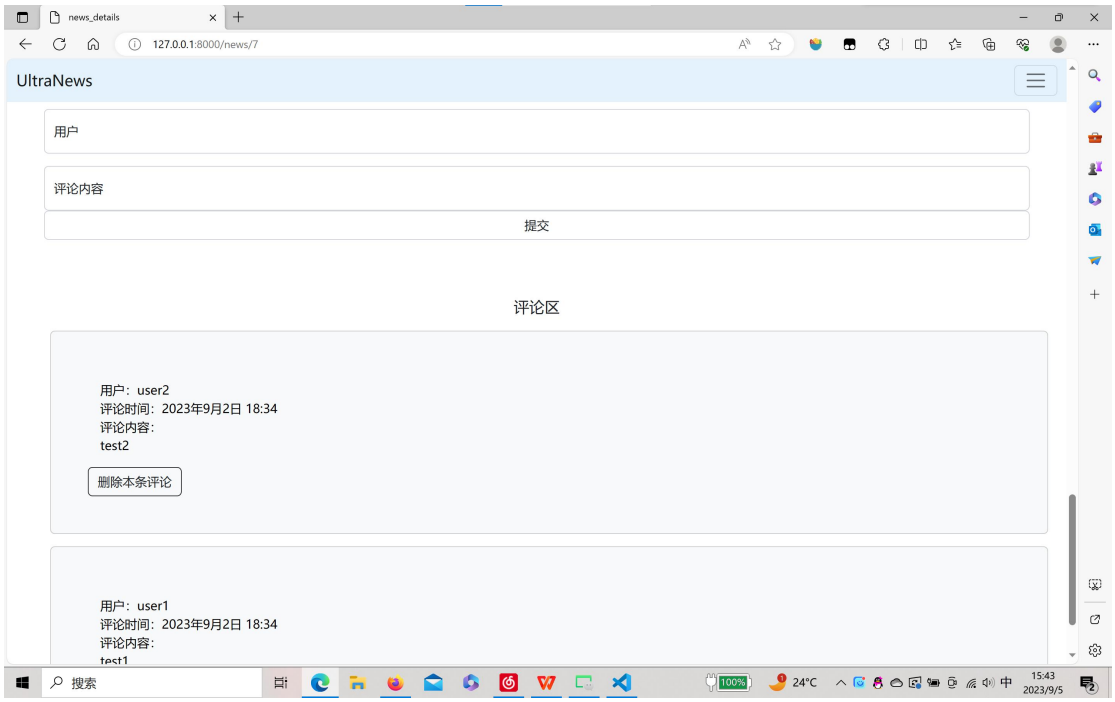


新闻列表页展示

新闻正文页含有这则新闻被爬取到的所有信息，网页展示的图片来自本地文件。新闻标题下面附有原新闻链接，点击即可跳转到原新闻的网页位置。页面下方含有文本输入框和按钮，用户可输入自己的用户名和评论内容对新闻进行评论。评论的文本以列表的形式显示在新闻页底部，且持久保留（即退出当前页面后再回来，评论仍在），评论条目按时间倒序排列。用户还可以按删除评论按钮删除评论。用户评论后和删除评论后这则新闻的评论数量会实时更新并且显示出来。

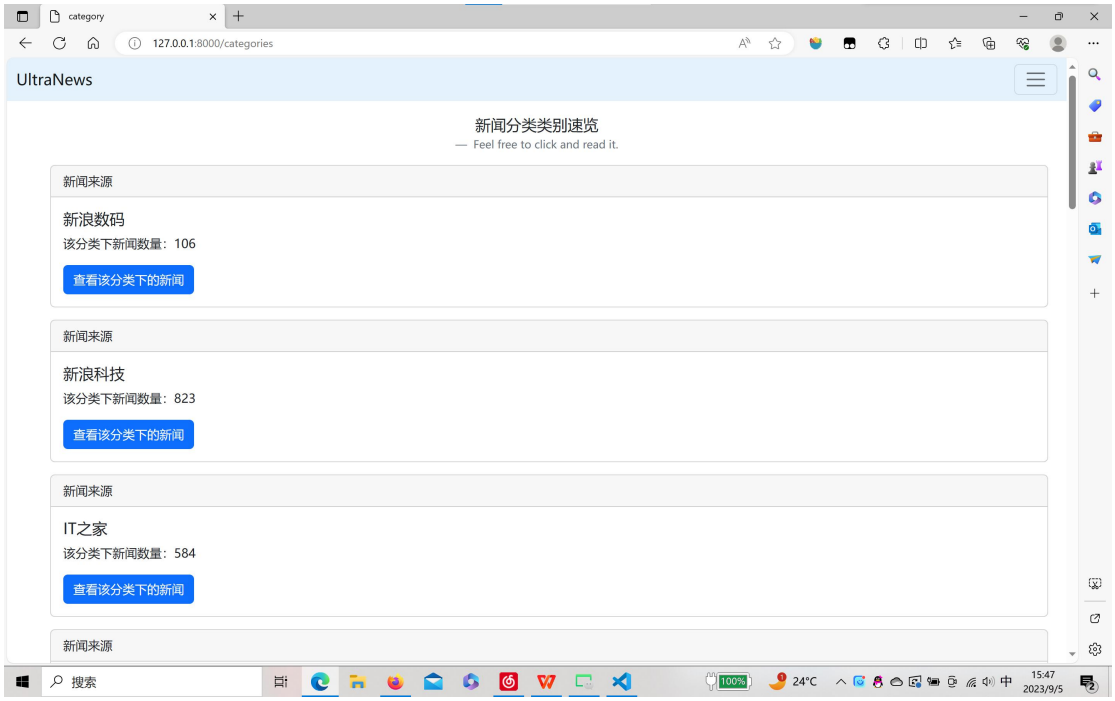


新闻正文页展示

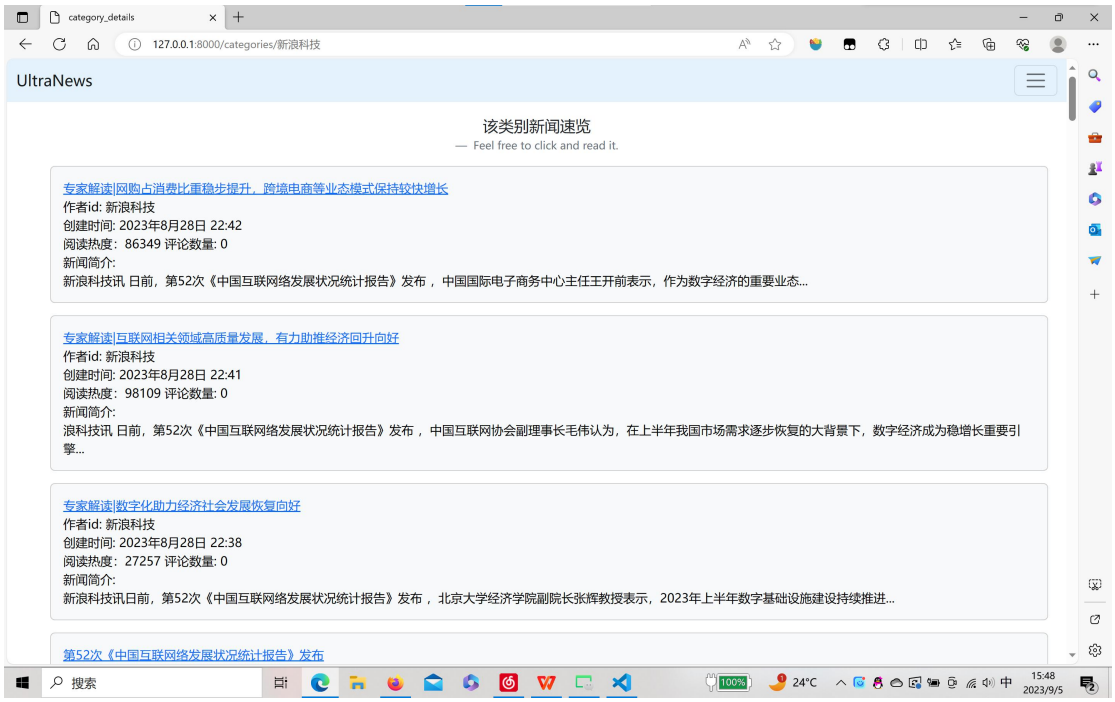


新闻正文页评论功能展示

分类页按照新闻来源将所有的新闻分成了 16 类，所有的类别均展示在分类页上，显示了分类名称与该分类下的新闻数量，点击按钮即可跳转到当前类别下的列表页。



新闻分类页展示



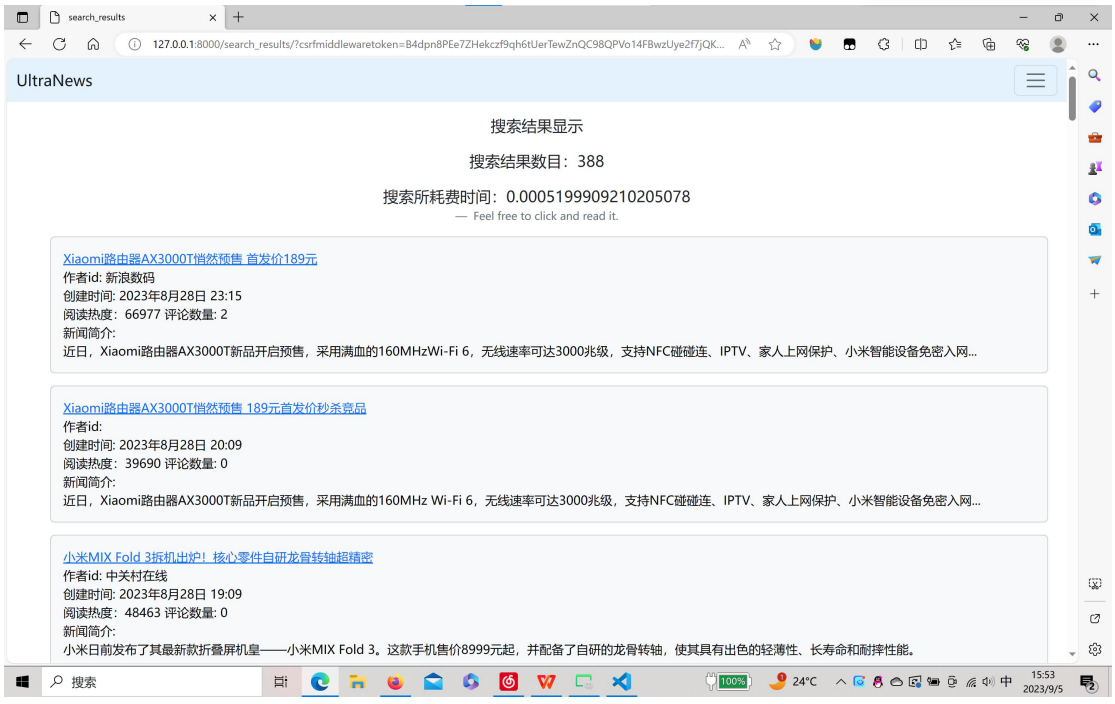
新闻分类详情页展示

回到首页的搜索功能，在搜索框里输入一段文本，会搜索标题或正文中包含该文本的新闻（仅考虑精确搜索）。此外提供了单选框和多选框，单选框可以按时间排序（由新到旧）和按热度排序（由高到低）对搜索结果进行排序，默认按时间进行排序。而多选框则是选中了需要搜索的新闻类别，无选项被勾选与所有选项被勾选均视为搜索所有分类下的新闻，默认设置为无选项被勾选。

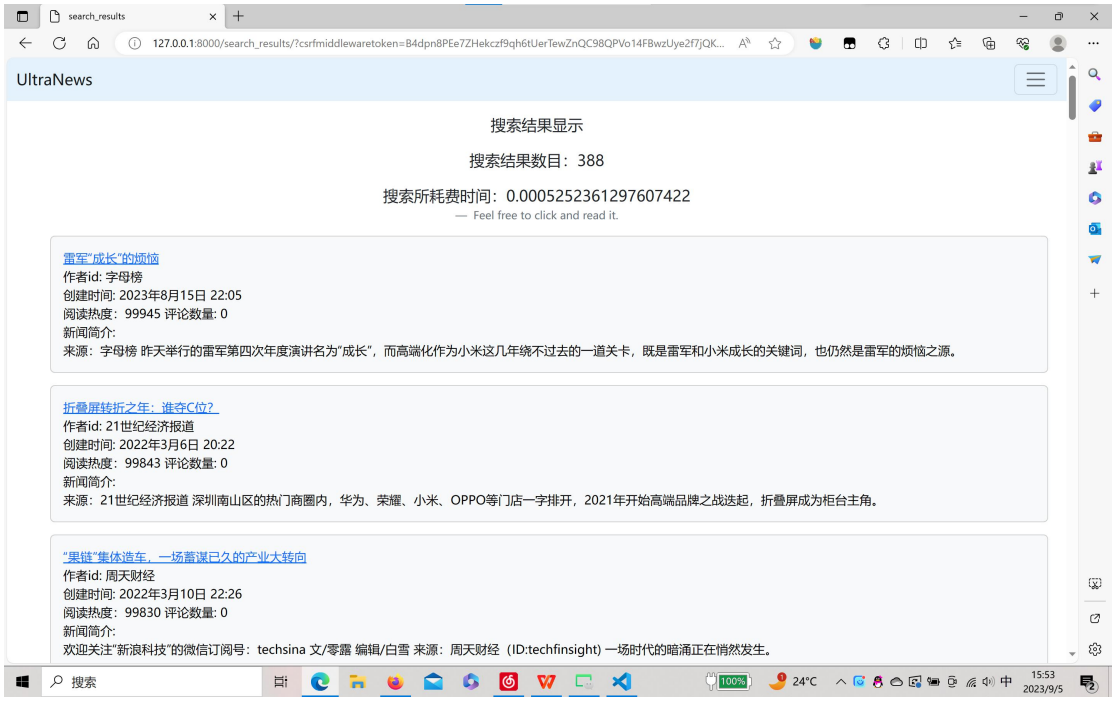
点击搜索按钮后会跳转到搜索结果页。搜索结果页最上方显示了搜索到的新闻数量和搜索时间，以分页的形式展示了搜索到的所有结果，同样的，点击每个条目会跳转到相应的信息页。如果未搜索到相关新闻，页面会显示“未搜索到此类结果。”

搜索功能通过对数据库的数据进行过滤来实现，主要使用 filter 过滤器。

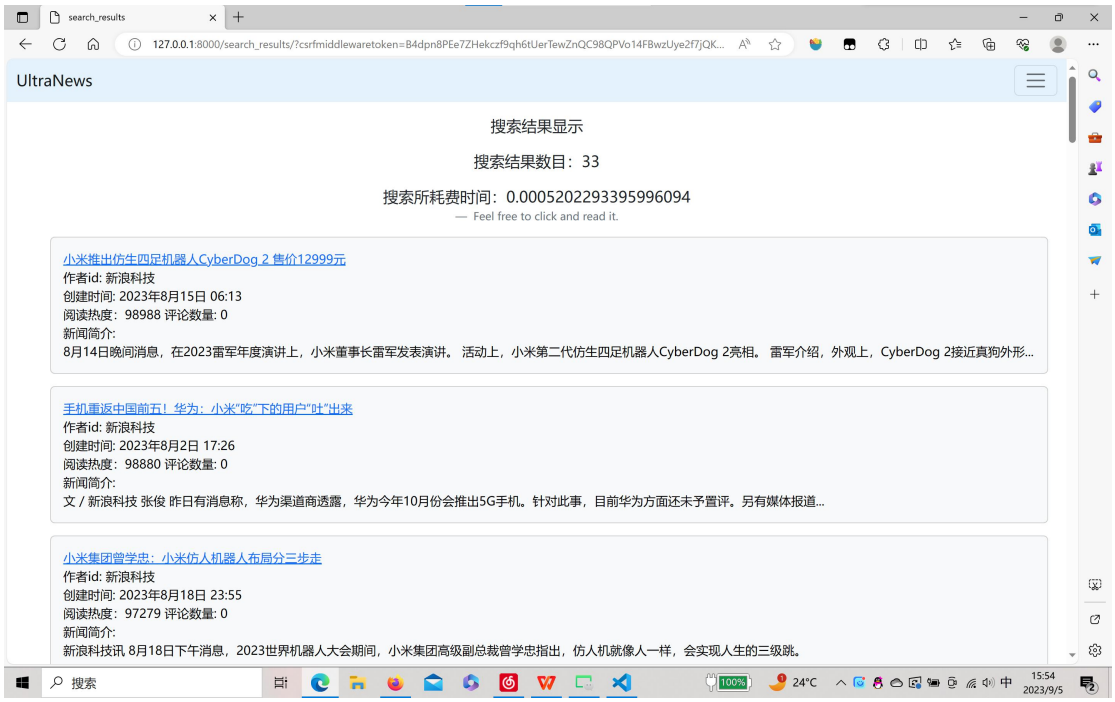
搜索功能展示



搜索结果按时间排序



搜索结果按热度排序



搜索结果按热度排序且选择分类类别的前两项

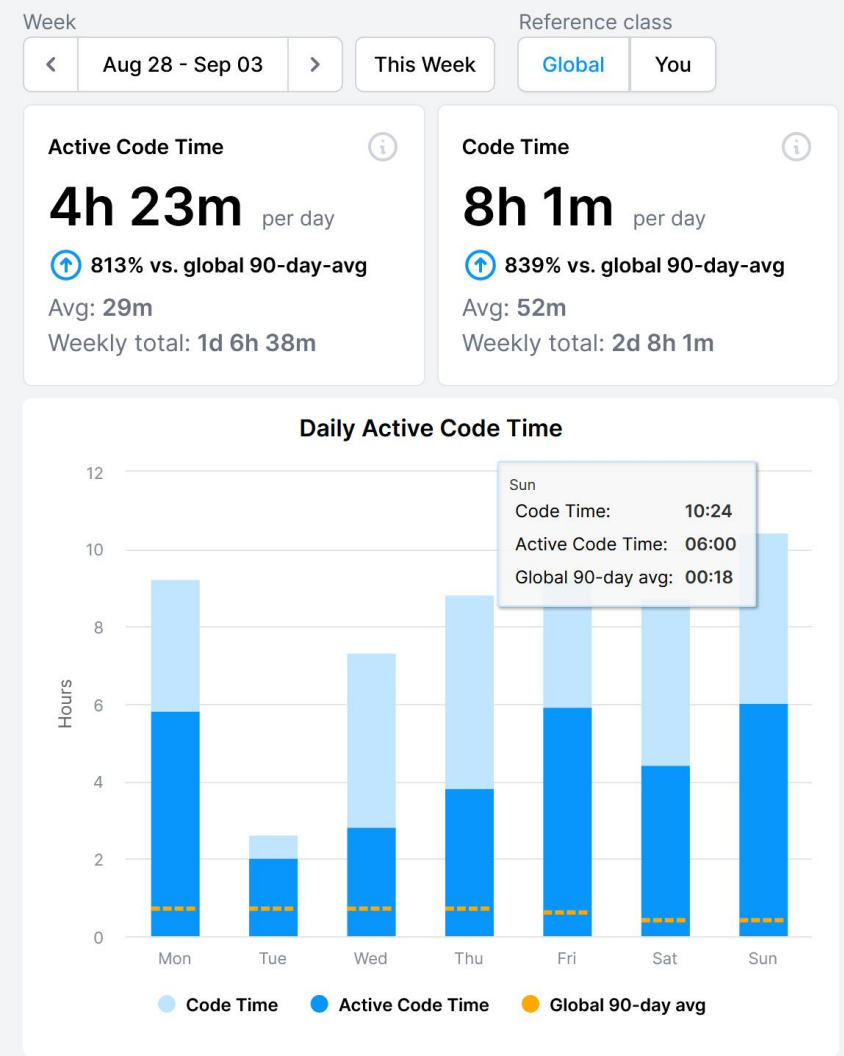
三、数据分析

数据分析部分通过对所爬取的数据进行分析，得到了三个结论，且使用 matplotlib、wordcloud 等工具使数据可视化，详见《数据分析报告》。

四、实验花费时间及感想

实验的三个部分中 Web 的系统搭建与设计花的时间最多。爬虫部分大概花了两三天，数据分析和写报告花了两天，其余的时间都在写 Web 系统。

调取了 vscode 插件中的数据，看到的时候被吓了一跳😇



完成整个实验的部分可以说是痛并快乐着。以前虽然也写过大作业，但这是第一次在短时间内完成一个很完整且所有部分都要自己写的大作业。虽然感觉时间有点赶，且课堂上教的东西很少，大部分都要自己课下学习与摸索。这就导致我几乎整天都坐在图书馆里，但是完成的那一刻还是很有成就感的。