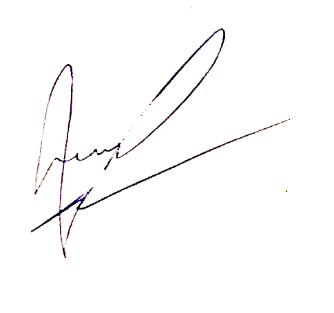
|  |  |  |
| --- | --- | --- |
| Optimized_image_813b5ec2.png | دانشگاه تهران  پردیس دانشکده­های فنی  دانشکده مهندسی برق و کامپیوتر | Images__Logo_FE.gif |
|  | | |
|  | | |
| راه‌اندازی و آماده‌سازی سامانه‌ی پویش و پردازش داده‌های بورس تهران  پايان‌نامه براي دريافت درجه کارشناسی  در رشته مهندسی کامپیوتر گرايش فناوری اطلاعات | | |
| **نام**  **علی قانع**  **شماره دانشجویی**  810194379 | | |
| **استاد راهنما:**  **دکتر هشام فیلی** | | |
|  | | |
| **شهریورماه ۱۴۰۰** | | |
|  | | |
|  | | |



|  |
| --- |
| **تعهدنامه اصالت اثر** |
| **باسمه تعالي** |
| **اينجانب علی قانع تائيد مي كنم كه مطالب مندرج در اين پایان نامه حاصل كار پژوهشي اينجانب است و به دستاوردهاي پژوهشي ديگران كه در اين نوشته از آنها استفاده شده است مطابق مقررات ارجاع گرديده است. اين پایان نامه قبلاٌ براي احراز هيچ مدرك هم سطح يا بالاتر ارائه نشده است.**  **كليه حقوق مادي و معنوي اين اثر متعلق به دانشكده فني دانشگاه تهران مي باشد.**  **نام و نام خانوادگي دانشجو : علی قانع**  **امضاي دانشجو :** |



**چکيده**

در سال‌های اخیر، با تغییرات اقتصادی به‌وجود آمده و ورود شرکت‌های بزرگ، بازار بورس رونق دوچندانی یافته و از استقبال عمومی بسیاری برخوردار گشته است. هر چند به دلیل عدم دانش فنی مناسب کاربران و پیچیدگی تحلیل‌های بورسی، امکان مبادله و سوددهی برای همه‌ی کاربران مهیا نیست اما این مورد از استقبال عمومی و سرمایه‌گذاری در بورس نکاسته است. از این رو نیاز به سامانه‌های گزارش و تحلیل بورس بیش از پیش احساس می شود که می‌توان گفت بستر کنونی وب‌سایت مدیریت فناوری بورس تهران[[1]](#footnote-1)، از ارائه‌ی آن عاجز است. اطلاعات مؤثر این بازار عمدتا به صورت دسته‌بندی‌نشده و غیرقابل تحلیل و تنها به صورت داده‌ی خام از طریق وب‌سایت قابل دسترسی است. پیش بینی قیمت و ارزش سهام یکی از مسائل مهمی است که در صورت امکان استفاده از آن می‌توان تصمیم بر خرید، فروش و یا حفظ یک سهم کرد. هدف این پروژه آماده‌سازی و فراهم‌کردن بستر مناسبی برای استفاده از روش‌های مختلف تحلیل اطلاعات با استفاده از اطلاعات با ارزش گردآوری‌شده از بازار بورس تهران در ساختاری مناسب است. در اختیار داشتن این داده‌ها زمینه‌ی استفاده از انواع روش‌ها برای پیش‌بینی آینده‌ی بازار را میسر می‌سازد.

**کلمات کلیدی: جمع آوری داده‌های بورس تهران، مدل داده بورس، پیاده‌سازی خزنده**

**فهرست مطالب**

Contents

[فصل 1: مقدمه و بيان مساله 1](#_Toc84127712)

[1-1- مقدمه 2](#_Toc84127713)

[1-2- تاريخچه‌ای از موضوع تحقيق 2](#_Toc84127714)

[1-3- شرح مسئله تحقيق 2](#_Toc84127715)

[1-4- تعريف موضوع تحقيق 3](#_Toc84127716)

[1-5- اهداف و آرمان‌های کلی تحقيق 3](#_Toc84127717)

[1-6- روش انجام تحقیق 4](#_Toc84127718)

[1-7- ساختار پايان‌نامه 4](#_Toc84127719)

[فصل 2: مفاهيم اوليه و پیش زمینه‌ی خزنده و مدل داده 5](#_Toc84127720)

[2-1- بررسی ساختار سایت شرکت مدیریت فناوری بورس تهران 6](#_Toc84127721)

[2-2- ساختار کلی خزنده 7](#_Toc84127722)

[2-3- روش توسعه‌ی خزنده‌ی جدید 7](#_Toc84127723)

[2-4- خلاصه و جمع‌بندی 8](#_Toc84127724)

[فصل 3: ساختار خزنده، معماری اجزا و روش کارکرد 10](#_Toc84127725)

[3-1- مقدمه 11](#_Toc84127726)

[3-2- جداول اصلی پایگاه داده‌ی پروژه 12](#_Toc84127727)

[3-2-2- جدول Instrument 12](#_Toc84127728)

[3-2-3- جدول Shareholder 13](#_Toc84127729)

[3-2-4- جدول Share 13](#_Toc84127730)

[3-2-5- جدول DayPriceThreshold 14](#_Toc84127731)

[3-2-6- جدول DayTradeSummary 15](#_Toc84127732)

[3-2-7- جدول ClientTradeInfo 16](#_Toc84127733)

[3-2-8- جدول TableOfInterest10 16](#_Toc84127734)

[3-2-9- جدول Bourse\_ChangeCapital 17](#_Toc84127735)

[3-3- دریافت اطلاعات 18](#_Toc84127736)

[3-3-1- روش دریافت اطلاعات 19](#_Toc84127737)

[3-3-2- ثبت اطلاعات اولیه 19](#_Toc84127738)

[3-3-3- ثبت سایر اطلاعات نمادها 20](#_Toc84127739)

[3-3-3-1- تابع Get\_ShareHolders 21](#_Toc84127740)

[3-3-3-2- تابع Get\_day\_price\_threshold 21](#_Toc84127741)

[3-3-3-3- تابع Get\_day\_trade\_summaries 21](#_Toc84127742)

[3-3-3-4- تابع Get\_client\_trade\_info 21](#_Toc84127743)

[3-3-3-5- تابع Create\_table\_of\_interest 22](#_Toc84127744)

[3-3-4- افزایش سرمایه‌ها 22](#_Toc84127745)

[3-3-4-1- فرمت اطلاعات 22](#_Toc84127746)

[3-3-5- توابع بازخوانی دوره‌ای اطلاعات 24](#_Toc84127747)

[3-3-5-1- دریافت اطلاعات نمادها به صورت روزانه 24](#_Toc84127748)

[3-3-5-2- دریافت اطلاعات افزایش سرمایه و تعدیل قیمت‌ها 24](#_Toc84127749)

[3-4- ذخیره‌سازی اطلاعات 25](#_Toc84127750)

[3-5- خلاصه و جمع‌بندی 25](#_Toc84127751)

[فصل 4: راه‌اندازی و اجرا خزنده و رابط کاربری 26](#_Toc84127752)

[4-1- آماده‌سازی سیستم 27](#_Toc84127753)

[4-2- اجرای خزنده‌ی کدها 27](#_Toc84127754)

[4-3- اجرای توابع اصلی 27](#_Toc84127755)

[4-4- رابط گرافیکی 28](#_Toc84127756)

[4-4-2- هدر 29](#_Toc84127757)

[4-4-3- منوی کناری 29](#_Toc84127758)

[4-4-4- محتوای اصلی 30](#_Toc84127759)

[4-5- خلاصه و جمع‌بندی 30](#_Toc84127760)

[فصل 5: جمع‌بندی، نتيجه‌گيری و پيشنهادها 31](#_Toc84127761)

[5-1- جمع‌بندی 32](#_Toc84127762)

[5-2- دستاوردها 32](#_Toc84127763)

[5-3- پیشنهادها 32](#_Toc84127764)

[فصل 6: مراجع 33](#_Toc84127765)

فهرست شکل­ها

[شکل (2-1) بخشی از صفحه‌ی اصلی نماد شستا در tsetmc.com 6](#_Toc82775975)

[شکل (2-2) بخشی از صفحه‌ی تاریخچه‌ی نماد شستا در cdn.tsetmc.com 6](#_Toc82775976)

[شکل (2-3) بخشی از صفحه‌ی اطلاعات نمادها 7](#_Toc82775977)

[شکل (2-4) نمونه‌ای از پاسخ APIها 8](#_Toc82775978)

[شکل (3-1) مدل داده‌ای قبلی پایگاه داده 11](#_Toc82775979)

[شکل (3-2) مدل داده‌ای پایگاه داده‌ی جدید 12](#_Toc82775980)

[شکل (3-3) شمای جدول نماد 13](#_Toc82775981)

[شکل (3-4) نمونه‌ی داده‌های جدول نماد 13](#_Toc82775982)

[شکل (3-5) شمای جدول سهام‌داران 13](#_Toc82775983)

[شکل (3-6) شمای جدول سهام 14](#_Toc82775984)

[شکل (3-7) نمونه‌ی داده‌های جدول سهام 14](#_Toc82775985)

[شکل (3-8) شمای جدول قیمت اطلاعات روزانه‌ی نماد 14](#_Toc82775986)

[شکل (3-9) نمونه‌ی داده‌های جدول قیمت اطلاعات روزانه‌ی نماد 15](#_Toc82775987)

[شکل (3-10) شمای جدول مبادلات روز نماد 15](#_Toc82775988)

[شکل (3-11) نمونه‌ی داده‌های جدول مبادلات روز نماد 16](#_Toc82775989)

[شکل (3-12) نمونه‌ی داده‌های جدول مبادلات روز نماد 16](#_Toc82775990)

[شکل (3-13) شمای جدول اطلاعات میانگین ده‌روزه‌ی معاملات 17](#_Toc82775991)

[شکل (3-14) شمای جدول اطلاعات افزایش سرمایه 18](#_Toc82775992)

[شکل (3-15) نمونه‌ی داده‌های جدول اطلاعات افزایش سرمایه 18](#_Toc82775993)

[شکل (3-16) خروجی API اطلاعات اولیه‌ی نماد به صورت JSON 20](#_Toc82775994)

[شکل (3-17) صفحه‌ی افزایش سرمایه‌ها 23](#_Toc82775995)

[شکل (3-18) اطلاعات صفحه‌ی افزایش سرمایه‌ها 23](#_Toc82775996)

[شکل (4-1) پیاده‌سازی رابط گرافیکی 29](#_Toc82775997)

1. مقدمه و بيان مساله

در این فصل نخست به تعریف و شرح مسأله و روش کلی تحقیق پرداخته، سپس مساله و موضوع مورد بررسی در این پایان‌نامه و اهداف و آرمان‌های کلی تحقیق بیان می‌شود و در انتها ساختار پایان‌نامه‌ی پیش رو ذکر شده است.

* 1. مقدمه

امروزه روش‌های تحلیل داده‌های بورسی بسیار گسترده‌اند و بهره‌گیری از این روش‌ها می‌تواند تضمینی برای سرمایه‌گذاری مطمئن و مبتنی بر اطلاعات باشد. در این میان اهمیت پیوستگی و دقت داده‌ها غیرقابل چشم‌پوشی و نیازمند جمع‌آوری و دسته‌بندی داده‌ها به سرعت و همراه با پیوستگی است. وجود یک بانک اطلاعاتی جامع و قدرت‌مند می تواند ابزاری قابل اتکا برای سرمایه‌گذاران این حوزه باشد و به کاربران در سرمایه‌گذاری مناسب کمک شایانی کند.

* 1. تاريخچه‌ای از موضوع تحقيق

این پروژه در امتداد پروژه‌ی «ایجاد مدل داده برای نمادهای بورس تهران به منظور پیش‌بینی قیمت سهام» که توسط سرکار خانم مریم کریمی زیر نظر دکتر هشام فیلی در مهرماه ۱۳۹۹ در دانشکده‌ی برق و کامپیوتر دانشگاه تهران ارائه شده، نوشته شده‌است. از این رو لازم به ذکر است قسمتی از مدل‌سازی‌ها و ساختار داده‌ای بر اساس این تحقیق طراحی و اجرا شده که به تفصیل در فصول آتی به شرح آن پرداخته خواهد شد.

* 1. شرح مسئله تحقيق

پیش‌بینی بازار بورس به معنی سعی در تشخیص ارزش سهام و موارد مورد مبادله در گذر زمان است. این تلاش، در جهت به‌دست‌آوردن سود از طریق خرید و فروش در این بازار صورت می‌گیرد. این موضوع از ابتدای پیدایش بازار بورس مورد توجه افراد و سرمایه‌گذاران مختلف بوده و هم‌اکنون نیز، توجه ویژه‌ای به آن می‌شود. برای پیش‌بینی بازار بورس روش‌های مختلفی به‌کار گرفته شده است. با رونق و پیش‌رفت دنیای اطلاعات در عصر نوین نیز این روش‌ها، که بیش‌تر به تحلیل‌های بورسی شناخته می‌شوند، از تأثیر فناوری مصون نمانده‌اند. از این رو نیاز به بستر مناسبی برای استفاده از روش‌های مختلف تحلیل اطلاعات با استفاده از اطلاعات باارزش گردآوری‌شده از بازار بیش از پیش احساس می شود.

* 1. تعريف موضوع تحقيق

در این پروژه، هدف فراهم‌سازی بستر ابتدایی برای استفاده از روش‌های تحلیل آماری و کامپیوتری با جمع‌آوری داده‌های پراهمیت بازار بورس تهران در ساختاری مناسب است. برای این منظور نیازمند پیاده‌سازی یک خزنده**[[2]](#footnote-2)** جهت جمع‌آوری داده‌های مورد نیاز است که طرح اولیه‌ی این خزنده توسط خانم مریم کریمی طراحی و پیاده‌سازی شده است. یکی از مسائل مهم در این خزنده، دسترسی سریع و دقیق اطلاعات در مقابل چالش‌های مختلفی است که برای آن وجود دارد. با استفاده از این خزنده می‌توان داده‌های مورد نیاز را جمع‌آوری کرده و با استفاده از آن‌ها امکان بهره‌گیری از روش‌های مختلف تحلیل برای پیش‌بینی بازار را فراهم آورد. مدل داده‌ای این خزنده و نحوه‌ی ذخیره‌سازی اطلاعات مؤثر و پراستفاده، دقت داده‌ای و کاربردی بودن آن می‌تواند کمک شایانی در کاربردپذیری این خزنده داشته باشد. در این رابطه بررسی و استقرار خزنده‌‌های قبلی و راه‌اندازی سامانه‌ی پویش و ذخیره‌سازی اطلاعات دریافتی با چالش‌های غیرقابل پیش‌بینی مختلفی مواجه بوده که یکی از اهداف اساسی این پروژه، رفع این چالش‌ها است. از سمت دیگر، ایجاد نمایشی مناسب و کاربرد‌پذیر نیازمند طراحی و توسعه‌ی رابط کاربری گرافیکی مناسب با امکانات مختلف است.

* 1. اهداف و آرمان‌های کلی تحقيق

اهداف کلی این پروژه شامل رفع ایرادات و ایجاد تغییرات ساختاری مورد نیاز روی خزنده‌های قبلی و به‌روزرسانی روش‌های واکشی اطلاعات است. هم‌چنین به تبع این تغییرات ساختاری، بازسازی مدل و ساختار پایگاه داده و ذخیره‌سازی داده‌ها نیز ضروری به‌نظر می‌رسد. با در دست‌داشتن اطلاعات پایه و تاریخچه‌ای و هم‌چنین روند مبادلات سهام، می‌توان از ابزار‌های قدرت‌مند تحلیل داده بهره گرفت و الگوهای متفاوتی را تشخیص داد.

* 1. روش انجام تحقیق

روش انجام این پروژه در گام‌های زیر خلاصه می‌شود:

* مطالعه‌ی اولیه روی مفاهیم اولیه‌ی بازار بورس
* کسب اطلاعات درباره‌ی نحوه‌ی پیاده‌سازی خزنده‌ی قبلی
* یافتن ایرادات خزنده‌ی قبلی و ایجاد خزنده‌ی جدید
* بازطراحی مدل داده‌ای و پایگاه داده
* راه‌اندازی سرور و استخراج منظم داده‌ها
* طراحی و پیاده‌سازی رابط گرافیکی و استقرار آن
  1. ساختار پايان‌نامه

فصل دوم، شامل بررسی تعاريف اساسی مربوط به حوزه‌ی بورس ، مفاهيم اوليه و اجزای اساسی خزنده و مدل داده، مروری بر پيشينه‌ی تحقيق و پيش‌زمينه‌های مورد نياز برای درک هرچه بهتر ساختار آن است.

فصل سوم در برگيرنده‌ی توضیحات مربوط به ساختار خزنده، معماری و اجزای آن و روش کارکرد آن است.

فصل چهارم به دستورالعمل اجرای مجدد پروژه روی سرور آماده‌سازی‌شده و یا روی سرور جدید اختصاص یافته است. هم‌چنین در این فصل به توضیحاتی در مورد رابط کاربری نیز خواهیم پرداخت.

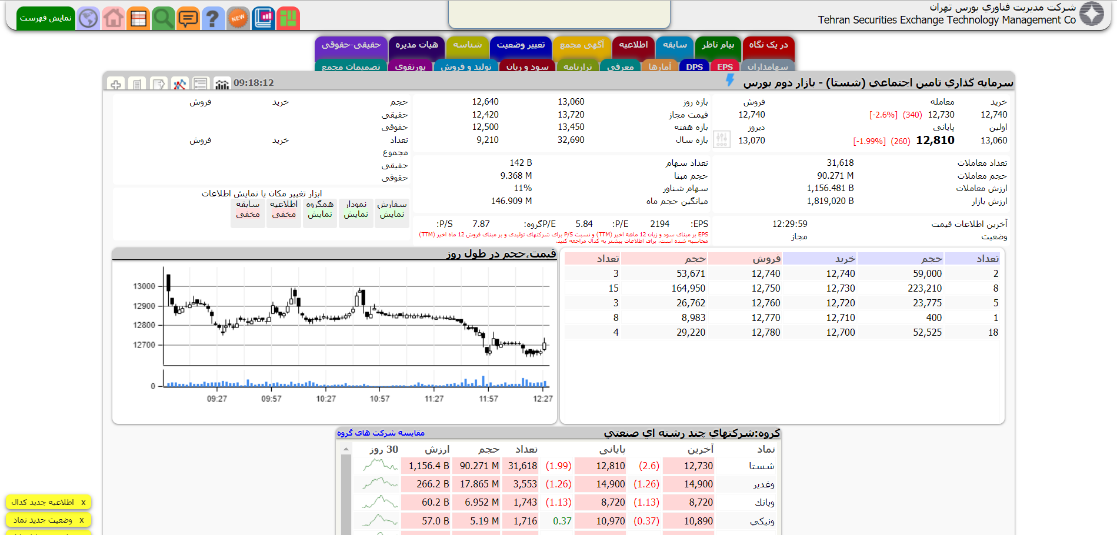
در نهايت، در فصل پنجم، نتيجه‌گيری‌های کلی حاصل‌شده در اين پروژه مورد بحث قرار می‌گیرد و پيشنهادهایی برای ادامه‌ی مسير به علاقمندان اين حوزه ارائه خواهد شد.

1. مفاهيم اوليه و پیش زمینه‌ی خزنده و مدل داده

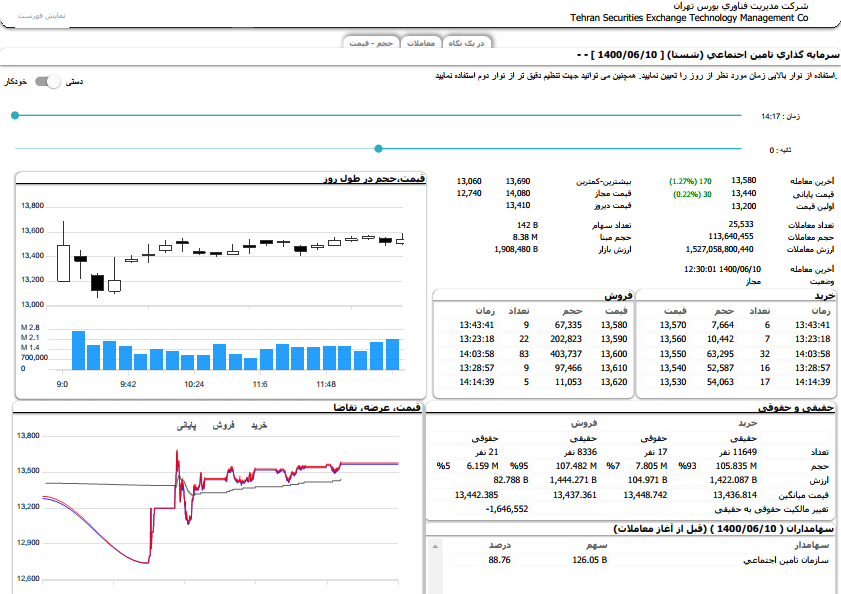
فصل پیش رو شامل تعاریف اساسی مربوط به حوزه‌ی بورس، مفاهیم اولیه و اجزای اساسی خزنده و مروری بر خزنده‌ی قبلی است.

بررسی ساختار سایت شرکت مدیریت فناوری بورس تهران

*تمامی اطلاعات مربوط به بازار بورس تهران در وب‌سایت به آدرس* tsetmc.com *توسط شرکت مدیریت فناوری بورس تهران در اختیار عموم کاربران قرار می‌گیرد. این وب‌سا‌یت اطلاعات مربوط به تمامی نمادها، معاملات انجام‌شده، عرضه و تقاضا، سهام‌داران و خریداران حقیقی و حقوقی را شامل می‌شود.(شکل ۲-۱) هم‌چنین تاریخچه‌ی تمامی رویدادهای مربوط به هر روز از هر سهام در صفحه‌ای اختصاصی در* cdn.tsetmc.com *قابل دسترسی است.(شکل ۲-۲)*

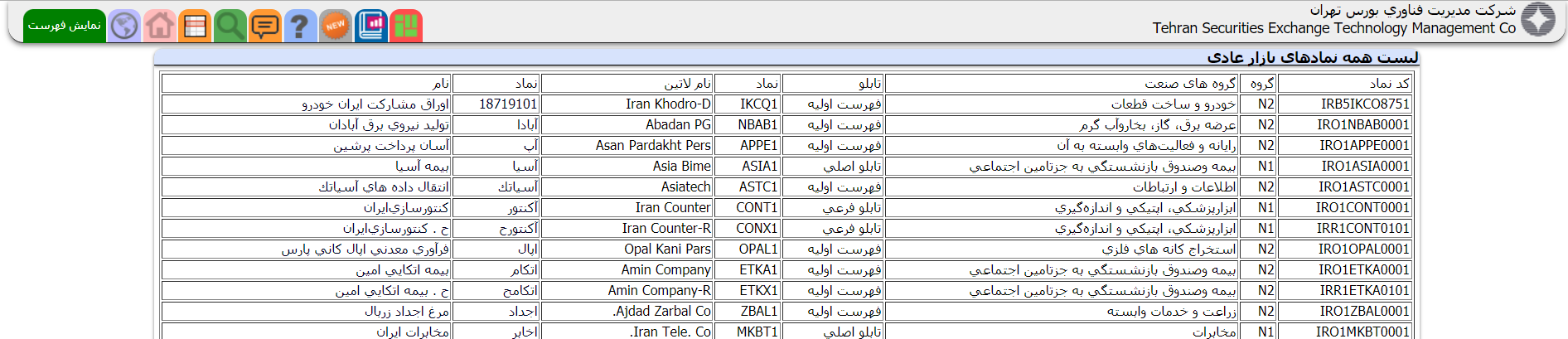
**

بخشی از صفحه‌ی اصلی نماد شستا در tsetmc.com



بخشی از صفحه‌ی تاریخچه‌ی نماد شستا در cdn.tsetmc.com

برای تمامی نماد‌ها یک کد منحصر به فرد وجود دارد که در تمامی لینک‌های دسترسی به آن نماد از آن کد استفاده می‌شود. لیستی از تمامی کدهای نمادها در سایت مدیریت فناوری بورس تهران موجود است که از طریق خواندن این لیست**[[3]](#footnote-3)** می‌توان تمامی کدها را دریافت کرد. (شکل ۲-۳)



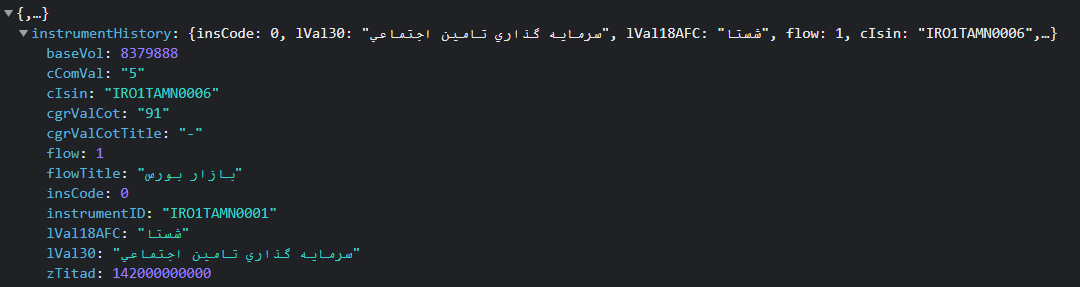
بخشی از صفحه‌ی اطلاعات نمادها

* 1. ساختار کلی خزنده

در پروژه‌ی «ایجاد مدل داده برای نمادهای بورس تهران به منظور پیش‌بینی قیمت سهام» طراحی و اجرا شده توسط سرکار خانم مریم کریمی زیر نظر دکتر هشام فیلی، از یک خزنده با فریم‌ورک Scrapy استفاده شده است. «اسکرپی» فریم‌ورکی از پایتون برای خزش در مقیاس بالا روی داده‌های وب است. ساختار این خزنده مبتنی بر خواندن اطلاعات از طریق تگ‌های HTML در صفحات وب است. در این پروژه با توجه به اصلاحات ساختاری صورت‌گرفته در وب‌سایت مدیریت فناوری بورس تهران، ناچار به اصلاحات ساختاری روی خزنده شدیم که در نهایت به طراحی و توسعه‌ی مجدد خزنده از ابتدا انجامید.

* 1. روش توسعه‌ی خزنده‌ی جدید

در میانه‌ی انجام این پروژه، اصلاحات ساختاری روی وب‌سایت مدیریت فناوری بورس تهران صورت گرفت. پیش از این اطلاعات نمادها توسط اسکریپتی درون بدنه‌ی HTML این صفحات بارگذاری می‌شد اما طی تغییرات اخیر، این اسکریپت جای خود را به APIهایی**[[4]](#footnote-4)** داده است که در این پروژه از آن‌ها بهره‌گیری می‌شود. (شکل ۲-۴)



نمونه‌ای از پاسخ APIها

همان‌طور که در شکل ۲-۴ مشاهده می‌شود، پاسخ تمامی APIها به صورت یک فایل JSON**[[5]](#footnote-5)** است. بدین ترتیب با ارسال درخواست به هر کدام از این APIها یک فایل JSON دریافت و واکشی اطلاعات از طریق آن انجام‌شده و به پایگاه داده منتقل می‌شود. یک فایل JSON شامل جفت‌های ویژگی-مقدار**[[6]](#footnote-6)** است.

خزنده‌ی جدید تحت فریم‌ورک جنگو طراحی و توسعه یافته است. جنگو بر اساس معماری MVT بوده که مخفف عبارت مدل، نمایش، الگو است. وقتی درخواستی به یک سرور وب می‌رسد، ابتدا این درخواست به جنگو منتقل می‌شود و جنگو سعی می‌کند بفهمد چه چیزی درخواست می‌شود. سپس جنگو لیستی از الگوها را می‌گیرد و سعی می‌کند URL را با آنها تطبیق دهد. جنگو الگوها را از بالا به پایین بررسی می‌کند و اگر درخواست با الگوی خاصی مطابقت داشته باشد، درخواست را به عملکرد مرتبط منتقل می‌کند. هم‌چنین قابلیت ORM**[[7]](#footnote-7)** جنگو بستر را برای هماهنگ‌سازی با پایگاه داده‌ی MySQL فراهم می‌آورد تا حداقل تغییرات نسبت به پایگاه داده‌ی قبلی را داشته باشیم.

* 1. خلاصه و جمع‌بندی

در این فصل با مفاهیم اولیه و پیش‌زمینه‌های پروژه که برای ایجاد درکی مناسب برای فهم مدل داده‌ای و خزنده مورد نیاز است، آشنا شدیم؛ به بررسی ساختار سایت tsetmc پرداختیم و صفحاتی که اطلاعات مورد نیاز در آن‌ها قرار گرفته بود را شناختیم. و نهایتا با توجه به نیازمندی‌های ذکرشده و تغییرات صورت‌گرفته، پایگاه داده و فریم‌ورک مناسب برای استخراج داده‌ها را دریافتیم.

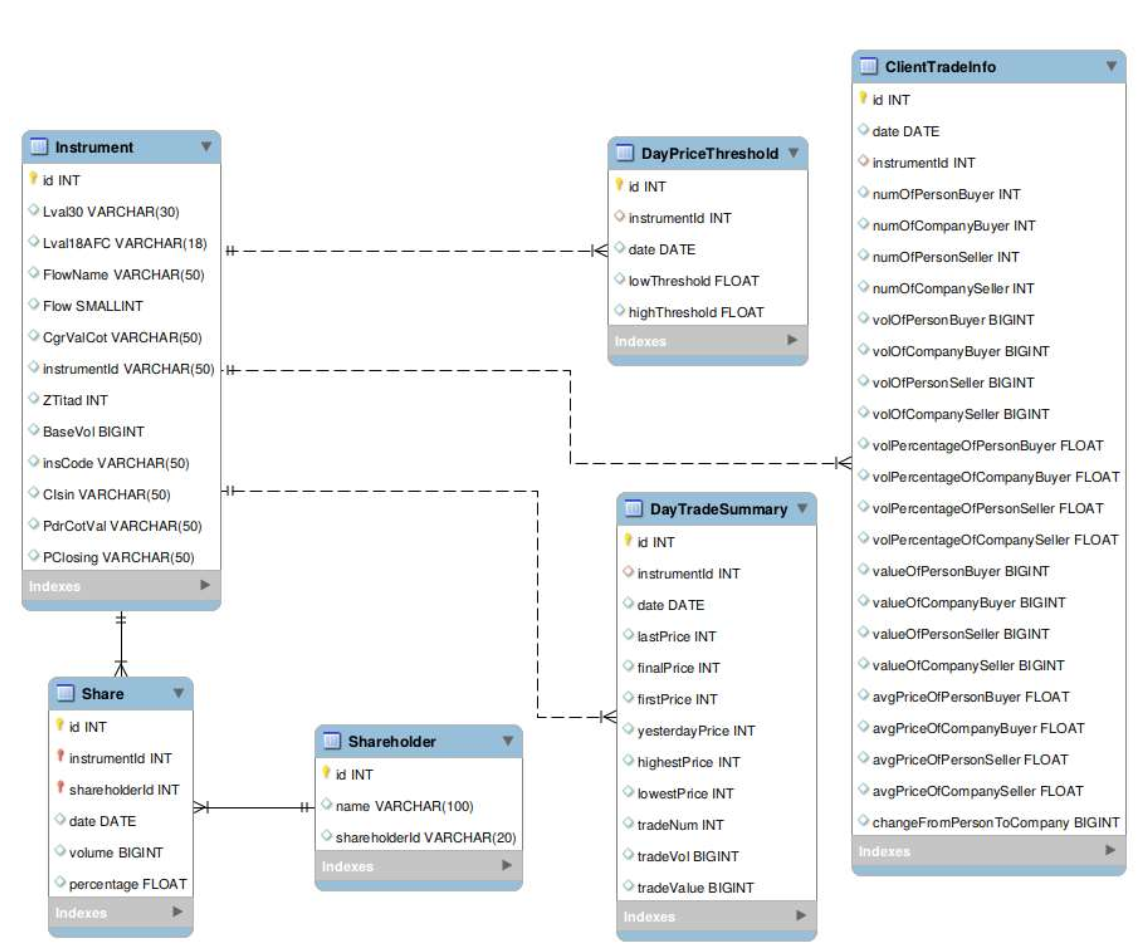
1. ساختار خزنده، معماری اجزا و روش کارکرد

فصل سوم در برگیرنده‌ی توضیحات مربوط به مدل داده‌ای پیشنهادی و ساختار خزنده مربوطه است.

* 1. مقدمه

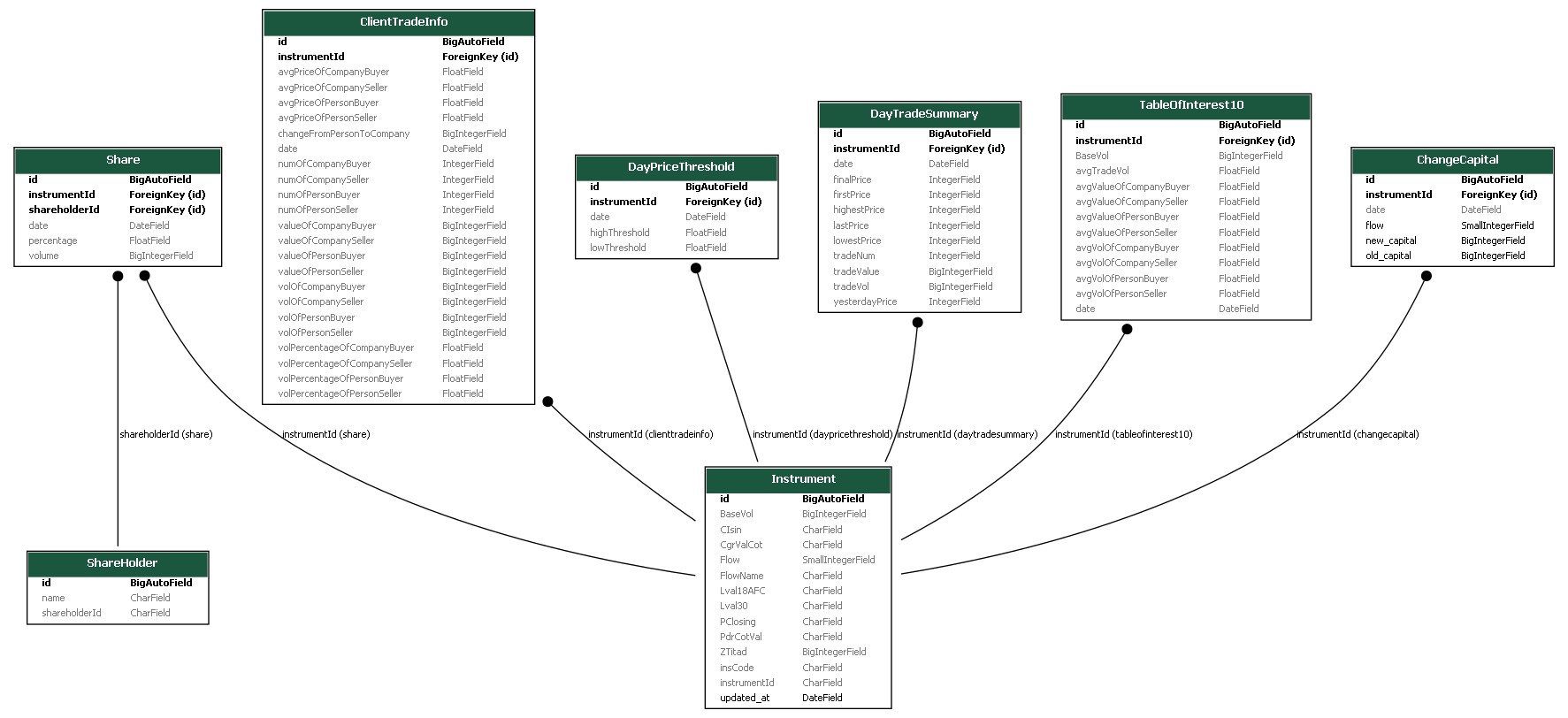
پایگاه داده‌ی پیاده‌سازی شده برای این پروژه از روی پایگاه داده‌ی قبلی(شکل ۳-۱ تهیه‌شده توسط خانم مریم کریمی) طراحی و بهینه‌سازی شده است؛ لیکن پایگاه داده‌ی جدید به دلیل استفاده از جنگو شامل جدوال پیش‌فرض این فریم‌ورک نیز هست. بنابراین می‌توان این پایگاه داده را به دو بخش تقسیم نمود؛ الف) جداول پیش‌فرض جنگو ب) جداول اصلی. در این گزارش از آوردن اطلاعات بخش «الف» چشم‌پوشی می‌کنیم و به توضیح بخش «ب» می‌پردازیم.

اطلاعات تمامی جداول به تفصیل در گزارش پایانی پروژه‌ی «ایجاد مدل داده برای نمادهای بورس تهران به منظور پیش‌بینی قیمت سهام» قابل دسترسی است و در این گزارش تنها به ذکر تغییرات به‌وجود‌آمده اکتفا می‌شود.



مدل داده‌ای قبلی پایگاه داده

* 1. جداول اصلی پایگاه داده‌ی پروژه

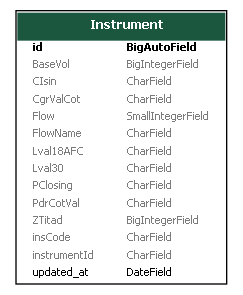


مدل داده‌ای پایگاه داده‌ی جدید

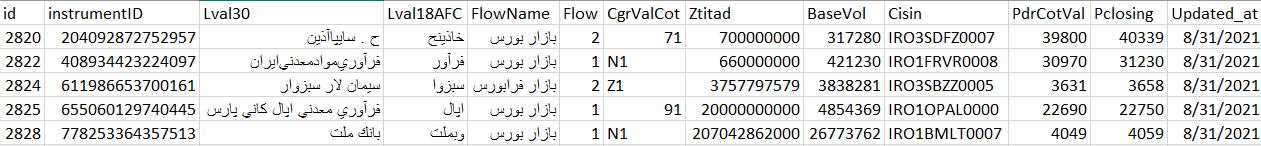
* + 1. جدول Instrument

این جدول اطلاعات نمادها را در خود نگه‌داری می‌کند. در واقع تمامی اطلاعات پایگاه داده از طریق دسترسی به این جدول و خواندن از روی آن تکمیل می‌شود. به دلیل استفاده از پیاده‌سازی قبلی، این جدول تنها جدولی است که اطلاعات آن در دو مرحله واکشی و تکمیل می‌شود. برای ساخت این جدول ابتدا با اجرای کد خزنده‌ی قبلی اطلاعات شماره‌ی نمادها باید در یک فایل با فرمت csv ذخیره‌سازی شود. البته می‌توان به صورت دستی نیز فایل را تکمیل و در اختیار خزنده گذاشت. این جدول در شکل ۳-۳ و ۳-۴ به نمایش گذاشته شده. تغییر به‌وجود‌آمده در این جدول تنها شامل اضافه‌شدن ستون زیر است:

Updated\_at : روزی که برای آن آخرین قیمت نماد ثبت شده. این ستون برای اطمینان از به‌روز بودن آخرین قیمت ضروری است.



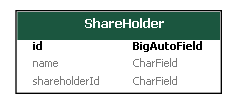
شمای جدول نماد



نمونه‌ی داده‌های جدول نماد

* + 1. جدول Shareholder

این جدول (شکل ۳-۵) شامل اطلاعات سهام‌داران تمامی نمادها و شامل اطلاعات زیر است :



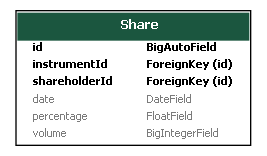
شمای جدول سهام‌داران

* + 1. جدول Share

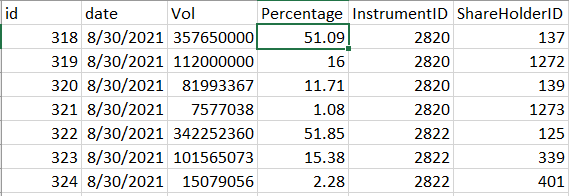
در این جدول (شکل۳-۶ و ۳-۷) اطلاعات مربوط به سهامی تمامی نمادها در هر روز نگهداری می‌شود. به ازای هر نماد در هر روز به تعداد سهام‌داران این نماد در جدول سطر وجود دارد. تغییرات این جدول نسبت به طراحی قبلی تنها شامل تغییر نام دو ستون زیر است:

* InstrumentID\_id: شناسه‌ی نماد.
* ShareholderID\_id: شناسه‌ی سهام‌دار.

لازم به ذکر است پسوند \_id در دو ستون آخر توسط خود فریم‌ورک جنگو برای کلیدهای خارجی اعمال می‌شود.



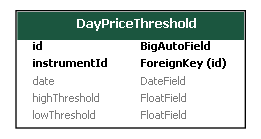
شمای جدول سهام



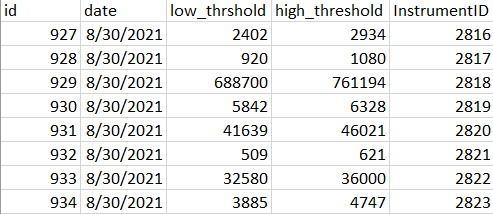
نمونه‌ی داده‌های جدول سهام

* + 1. جدول DayPriceThreshold

این جدول (شکل ۳-۸ و ۳-۹) شامل بازه‌ی قیمت معاملات هر روز برای هر نماد است.



شمای جدول قیمت اطلاعات روزانه‌ی نماد

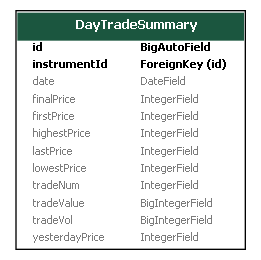


نمونه‌ی داده‌های جدول قیمت اطلاعات روزانه‌ی نماد

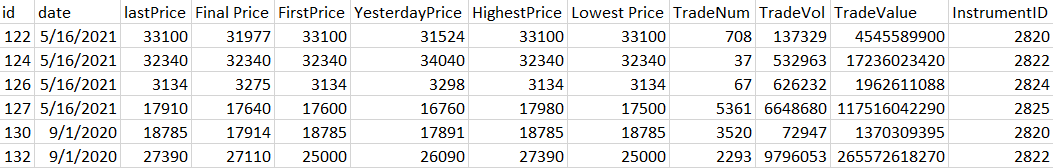
* + 1. جدول DayTradeSummary

این جدول (شکل ۳-۱۰ و ۳-۱۱) شامل اطلاعات مربوط به معاملات یک نماد برای یک روز مشخص است. با توجه به اضافه‌شدن اطلاعات افزایش سرمایه‌ی نمادها، ستون‌های زیر برای نگهداری از قیمت‌های تعدیل‌شده‌ی نمادها به این جدول اضافه‌شده اند:

* BalancedPrice: قیمت پایانی تعدیل‌شده برای نماد
* ‌BalancedHigh: مقدار تعدیل‌شده‌ی بالاترین قیمت نماد
* BalancedLow: مقدار تعدیل‌شده‌ی کم‌ترین قیمت نماد
* BalancedFirst: مقدار تعدیل‌شده‌ی قیمت بازشدن نماد



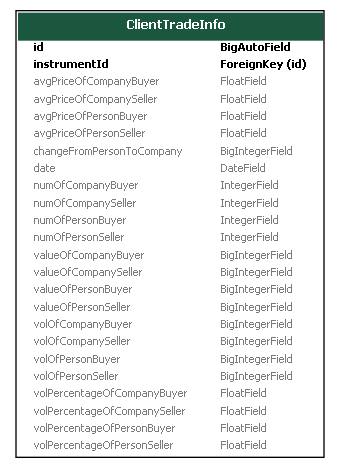
شمای جدول مبادلات روز نماد



نمونه‌ی داده‌های جدول مبادلات روز نماد

* + 1. جدول ClientTradeInfo

این جدول (شکل ۳-۱۲) مشروح رویدادهای خرید و فروش مشتریان حقیقی و حقوقی است.

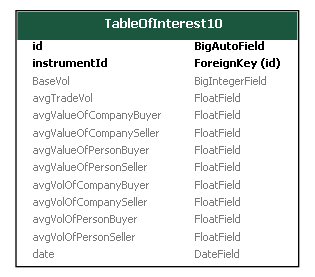


نمونه‌ی داده‌های جدول مبادلات روز نماد

* + 1. جدول TableOfInterest10

این جدول اطلاعات مربوط به میانگین‌های ده‌روزه‌ی معاملات هر سهام را شامل می‌شود. لازم به ذکر است این اطلاعات در بازه‌ی ده روز اول کراول قابل استناد نیست. (شکل ۳-۱۳)

* Id: ردیف هر سطر از اطلاعات
* Date: تاریخ میلادی که این اطلاعات مربوط به آن است.
* instrumentID\_id: شناسه‌ی نماد
* BaseVol: حجم مبنای نماد
* AvgTradeVol: میانگین ده‌روزه‌ی حجم کل معاملات
* AvgVolOfPersonBuyer: میانگین ده‌روزه‌ی حجم خرید خریداران حقیقی.
* AvgVolOfCompanyBuyer: میانگین ده‌روزه‌ی حجم خرید خریداران حقوقی.
* AvgVolOfPersonSeller: میانگین ده‌روزه‌ی حجم فروش فروشندگان حقیقی.
* AvgVolOfCompanySeller: میانگین ده‌روزه‌ی حجم فروش فروشندگان حقوقی.
* AvgValueOfPersonBuyer: میانگین ده‌روزه‌ی ارزش معاملات خریداران حقیقی.
* AvgValueOfCompanyBuyer: میانگین ده‌روزه‌ی ارزش معاملات خریداران حقوقی.
* AvgValueOfPersonSeller: میانگین ده‌روزه‌ی ارزش معاملات فروشندگان حقیقی.
* AvgValueOfCompanySeller: میانگین ده‌روزه‌ی ارزش معاملات فروشندگان حقوقی.

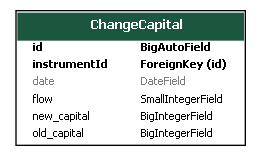


شمای جدول اطلاعات میانگین ده‌روزه‌ی معاملات

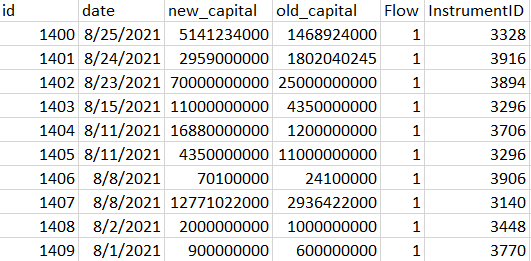
* + 1. جدول Bourse\_ChangeCapital

این جدول شامل تاریخچه و اطلاعات مربوط به افزایش سرمایه‌ی نمادها است (شکل ۳-۱۴ و ۳-۱۵)

* Id: ردیف هر سطر از اطلاعات
* Date : تاریخ میلادی که افزایش سرمایه در آن صورت گرفته است.
* instrumentID\_id : شناسه‌ی نماد
* Flow : کد بازاری است که نماد در آن قرار دارد و می‌تواند از این اعداد باشد : ۱- بورس ۲- فرابورس
* Old\_capital : میزان سرمایه‌ی قبلی نماد.
* New\_capital : میزان سرمایه‌ی جدید نماد.



شمای جدول اطلاعات افزایش سرمایه



نمونه‌ی داده‌های جدول اطلاعات افزایش سرمایه

* 1. دریافت اطلاعات

دریافت اطلاعات از طریق APIهایی که سایت tsetmc تعبیه کرده صورت می‌پذیرد. در سایت cdn.tsetmc.com برای هر نماد در هر روز صفحه‌ای وجود دارد که شامل اطلاعات آن نماد در آن روز است. پیش از این اطلاعات نمادها توسط اسکریپتی درون بدنه‌ی HTML این صفحه بارگذاری می‌شد اما طی تغییرات اخیر، این اسکریپت جای خود را به APIهایی داده که در این پروژه از آن‌ها بهره‌گیری می‌شود. لازم به ذکر است دلیل بازنویسی خزنده‌ی قبلی و هم‌چنین تغییر فریم‌ورک پایه از scrapy به django، همین تغییر رویه در وب‌سایت مدیریت فناوری بورس است. بدین ترتیب با ارسال درخواست به هر کدام از این APIها یک فایل Json دریافت و واکشی اطلاعات از طریق آن انجام‌ و به پایگاه داده منتقل می‌شود.

خزنده‌ی جدید شامل دوفایل instrument\_services و others است که فایل اول به واکشی اطلاعات نمادها پرداخته و باقی اطلاعات توسط فایل دوم استخراج می‌شود. دلیل آن که این خزنده به دو فایل تقسیم‌شده، تفاوت کاربردی این دو و استفاده از فایل instrument تنها یک بار در ابتدای واکشی اطلاعات است.

* + 1. روش دریافت اطلاعات

در cdn برای هر نماد در هر روز، فرمت لینک صفحه به شکل زیر است:

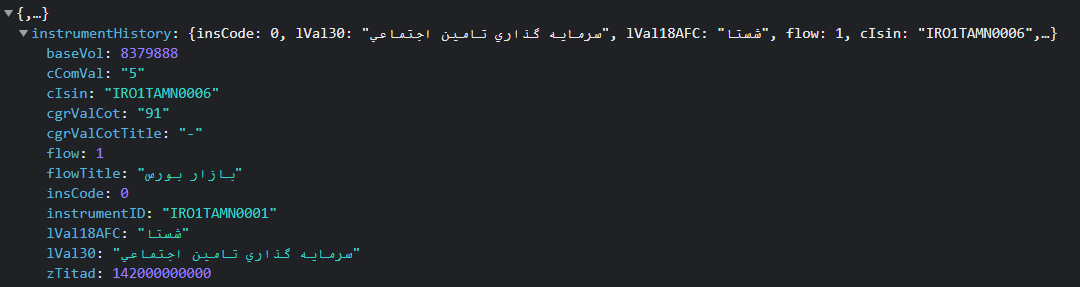
Cdn.tsetmc.com/history/[instrumentID]/[Date]

که در آن InstrumentID همان InstrumentID است که در فایل csv ذخیره شده و Date تاریخ میلادی به فرمت YMD است. از آن‌جا که اطلاعات دیگر مستقیما در فایل این صفحات بارگذاری نمی‌شود می‌توانیم از API برای دریافت اطلاعات از هر بخش استفاده کنیم.

* + 1. ثبت اطلاعات اولیه

در فایل instrument\_services دو تابع get\_instrument\_ids و get\_instruments وجود دارد. تابع get\_instrument\_ids از طریق فایل csv کد نماد را برای تابع دیگر واکشی می‌کند و در اختیار آن قرار می‌دهد. از طرف دیگر،‌ تابع get\_instruments به ازای هر خط از اطلاعات، یک بار از طریق API اطلاعات را واکشی و در دیتابیس قرار می‌دهد. این API از طریق لینک زیر قابل دسترسی است:

Cdn.tsetmc.com/api/Instrument/GetInstrumentHistory/[id]/[date]



خروجی API اطلاعات اولیه‌ی نماد به صورت JSON

طبیعتا برای خواندن اطلاعات نماد از به‌روزترین تاریخ ممکن، یعنی تاریخ روزی که در حال اجرای این تابع هستیم، استفاده می‌کنیم. بدین ترتیب با ارسال درخواست به این API داده‌های زیر استخراج و ثبت می‌گردد: نام ‌فارسی نماد، نام فارسی خلاصه‌شده‌ی نماد، نام و کد بازار نماد، کد گروه نماد، تعداد سهام، حجم مبنا، کد شرکت، آخرین قیمت و قیمت نهایی امروز.

مشخصا آخرین قیمت و قیمت نهایی یک سهم که در جدول Instrument وجود دارد، در طول زمان امکان تغییر دارد که در ادامه به آن نیز خواهیم پرداخت.

* + 1. ثبت سایر اطلاعات نمادها

در فایل others.py توابع دیگری با کارکردی بسیار شبیه به تابع get\_instruments در فایل instruments\_services وجود دارد که هر کدام یکی از جداول پایگاه داده را پر می‌کنند. در این فایل یک تابع period\_generator برای تولید تاریخ‌های مورد نیاز برای واکشی اطلاعات نیز وجود دارد. تابع period\_generator به ازای دو ورودی تاریخ‌های شروع و پایان، تمامی تاریخ‌های درون آن بازه را تولید و به تابع اصلی ارسال می‌کند. تابع اصلی ما در این فایل تابع crawl است که به ازای هر روزی که در period\_generator تولید می‌شود، تمامی توابع دیگر را برای آن تاریخ صدا می‌زند. این توابع، بر خلاف تابع کراول نمادها، در صورت تکرار کراول (هم‌پوشانی بازه‌ی تاریخی در دوبار اجرای خزنده) اطلاعات را دوباره تولید نکرده و در نتیجه مشکلی ایجاد نمی‌کنند. این توابع شامل موارد زیر است:

* + - 1. تابع Get\_ShareHolders

این تابع اطلاعات تمامی سهام‌داران تمامی نمادها را بررسی کرده و اگر اطلاعات سهام‌داری در پایگاه داده (جدول Shareholders) موجود نباشد، آن را ایجاد می‌کند. سپس اطلاعات مربوط به این سهام‌داران را در پایگاه داده(جدول share) ذخیره می‌نماید.

* cdn.tsetmc.com/api/shareholder/[intrumentID]/[date]
  + - 1. تابع Get\_day\_price\_threshold

‌ این تابع از طریق API تمامی اطلاعات مربوط به جدول DayPriceThreshold را واکشی و ذخیره می‌کنند.

* cdn.tsetmc.com/api/MarketData/GetStaticThreshold/[InstrumentID]/[Date]
  + - 1. تابع Get\_day\_trade\_summaries

این تابع ابتدا اطلاعات جدول dayTradeSummary را تکمیل کرده و بعد در صورتی که تاریخ تکمیل (ذخیره‌شده در ستون Updated\_at در جدول نماد) ستون‌های قیمت نهایی و آخرین قیمت جدول Instrument از این تاریخ قدیمی‌تر باشد، آن را نیز به روزرسانی می‌کند.

* cdn.tsetmc.com/API/ClosingPrice/GetClosingPriceDaily/[InstrumentID]/[Date]
  + - 1. تابع Get\_client\_trade\_info

این تابع مقادیر جدول ClientTradeInfo را واکشی و در پایگاه داده ذخیره‌سازی می‌کند.

* cdn.tsetmc.com/api/ClientType/GetClientTypeHistory/[InstrumentID]/[Date]
  + - 1. تابع Create\_table\_of\_interest

این تابع از طریق API عمل نکرده و با انجام محاسبه بر روی اطلاعات ده‌روز گذشته‌ی نماد در دو جدول DayTradeSummary و ClientTradeInfo اطلاعات جدول TableOfInterest10 را تولید می‌کند.

* + 1. افزایش سرمایه‌ها

دو تابع دیگر نیز در این فایل وجود دارند که مخصوص خواندن اطلاعات مربوط به افزایش سرمایه‌ها است. اطلاعات مربوط به افزایش سرمایه‌ها از ابتدا تا کنون در دو صفحه (تصویر ۳-۱۷) نگه‌داری می‌شود. آدرس این صفحات به شکل زیر است:

* Tsetmc.com/loader.aspx?partree=151310&flow=1
* Tsetmc.com/loader.aspx?partree=151310&flow=2

که اولی به بازار بورس و دومی به نمادهای فرابورس مربوط می‌شود. لازم به ذکر است سطر اول هر کدام از جداول این صفحات، بی‌استفاده است و باید از سطر دوم (ایندکس ۱) شروع به خواندن کنیم.

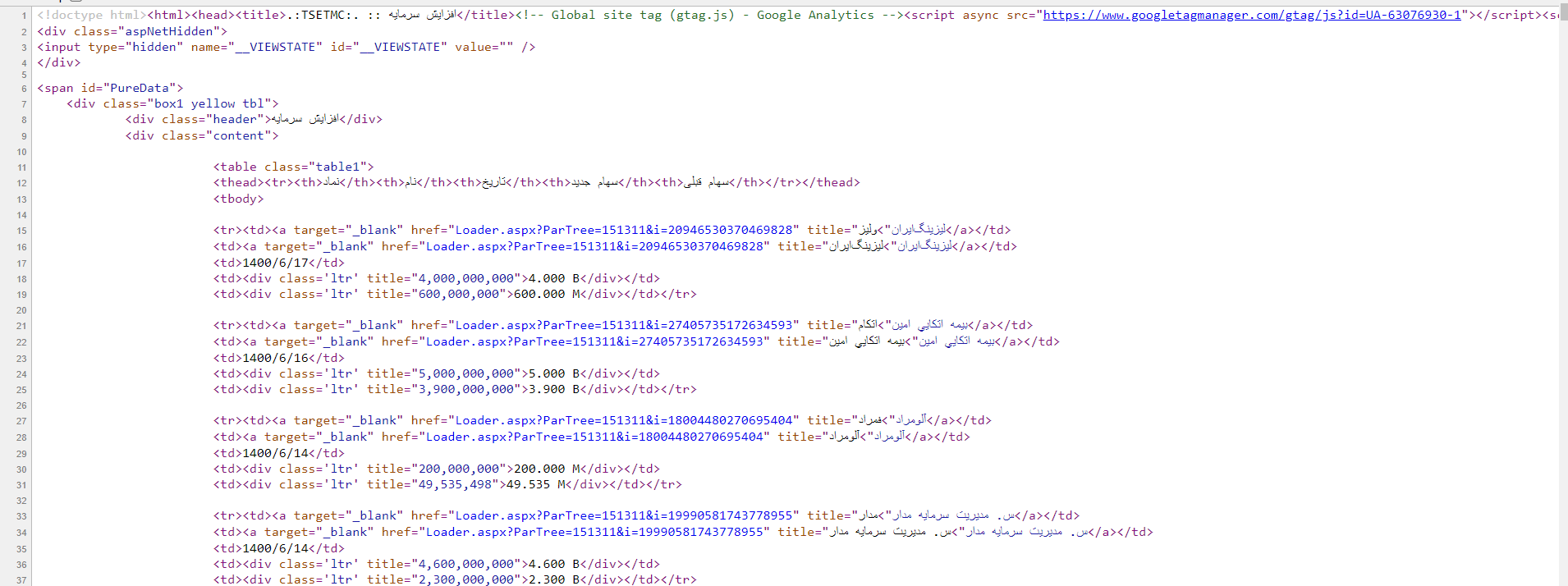
* + - 1. فرمت اطلاعات

اطلاعات مربوط به افزایش سرمایه‌ها در جدولی نگه‌داری می‌شوند که هر ردیف، مربوط به یک افزایش سرمایه است. با بررسی کد HTML صفحه (تصویر ۳-۱۸) در می‌یابیم که اطلاعات به صورت زیر نگه‌داری می‌شود:

* در هر ردیف کد نماد در انتهای href دیتای اول قرار دارد. دو td اول شامل نام فارسی و نام اختصاری فارسی نیز هست که می‌دانیم پیش از این در جدول نماد ذخیره شده و نیازی به واکشی آن‌ها نیست.
* تاریخ مربوط به این افزایش سرمایه درون td سوم به فرمت جلالی Y/M/D قرار دارد.
* عدد دقیق سرمایه‌ی جدید و سرمایه‌ی قبلی به ترتیب درون td چهارم و پنجم و به فرمت متنی با جداکننده‌ی سه‌رقم قرار گرفته است.



صفحه‌ی افزایش سرمایه‌ها



اطلاعات صفحه‌ی افزایش سرمایه‌ها

به این ترتیب برای تکمیل اطلاعات مربوط به افزایش سرمایه نیاز است تا برای هر flow:

* ابتدا کد HTML صفحه را خوانده و trها را جداسازی کنیم.
* برای هر tr از td اول کد نماد را استخراج کرده و شناسه‌ی نماد را به دست بیاوریم. با توجه به ثابت بودن طول href می‌دانیم کد نماد از ایندکس ۲۹ شروع می‌شود.
* سپس با استفاده از مبدل تاریخ جلالی به میلادی td سوم را به فرمت استاندارد پایگاه داده که میلادی است، تبدیل کنیم.
* سپس با استفاده از یک تابع کمکی(تابع get\_num\_from\_str ) اطلاعات متنی td چهارم و پنجم را به عدد تبدیل کنیم.
* در نهایت این اطلاعات را در پایگاه داده ذخیره کنیم.

کتاب‌خانه‌‌های استفاده‌شده برای این عملیات، BeautifulSoup و Jdatetime است که اولی برای خواندن اطلاعات HTML با استفاده از تگ‌ها و دومی برای تبدیل تاریخ جلالی به میلادی کاربرد دارد. با خواندن هر ردیف از جدول، اطلاعات یک افزایش سرمایه به دست می‌آید.

* + 1. توابع بازخوانی دوره‌ای اطلاعات

در انتها برای بازخوانی دوره‌ای اطلاعات، دو تابع job\_crawl و job\_capital به همراه توابع کمکی start\_crawling و start\_change\_capitals اجرا شده‌اند. این توابع با استفاده از توابعی که پیش از این شرح داده شده‌اند، به ذخیره‌سازی اطلاعات در پایگاه داده می‌پردازند. این توابع را می‌توان با استفاده از ساخت ترمینال‌های مجازی توسط screen همیشه در حال اجرا نگاه داشت.

* + - 1. دریافت اطلاعات نمادها به صورت روزانه

تابع job\_crawl در هر ساعت یک‌بار اجرا می‌شود و در ساعتی از شبانه‌روز که از قبل در آن تنظیم‌شده، با فراخوانی تابع start\_crawling اطلاعات دو روز قبل را واکشی و ذخیره‌سازی می‌کند. این رویکرد برای اطمینان از صحت دریافت اطلاعات پیاده‌سازی شده است. در صورتی که ساعت اجرا با ساعت مقرر یکسان نباشد، این تابع با استفاده از متد به‌خواب‌رفتن پایتون یک ساعت توقف می‌کند.

* + - 1. دریافت اطلاعات افزایش سرمایه و تعدیل قیمت‌ها

تابع job\_capital در هر روز یک‌بار اجرا می‌شود.(با استفاده از همان متد به‌خواب‌رفتن) این تابع در روز مشخصی از هفته، با اجرای تابع کمکی start\_change\_capitals ابتدا اطلاعات افزایش سرمایه را بازخوانی و سپس قیمت تعدیل‌شده‌ی تمامی نمادها را با توجه به افزایش سرمایه‌های جدید اصلاح می‌کند.

* 1. ذخیره‌سازی اطلاعات

اطلاعات در دیتابیسی با عنوان Tehran\_stock\_exchange ذخیره می‌شوند. برای تغییر این نام می‌توان در فایل settings.py نام دیتابیس را عوض کرد. هم‌چنین می‌توان با استفاده از تغییر فایل Models.py، تغییرات دلخواه را در ساختار مدل داده‌ای ایجاد نمود؛ لازم به ذکر است پس از انجام این عملیات، باید migrationهای لازم دوباره صورت پذیرد.

* 1. خلاصه و جمع‌بندی

فصل سوم به طور عمده شیوه‌ی عملی اجرای پروژه را در بر می‌گیرد که شامل مستندات پایگاه داده و پیاده‌سازی خزنده به همراه اطلاعات فنی پروژه است.

1. راه‌اندازی و اجرا خزنده و رابط کاربری

پس از توضیحات فنی و اجرایی، در این فصل در مورد نحوه‌ی راه‌اندازی و اجرای خزنده بر روی سرور میزبان و استفاده از آن بر روی سایر سیستم‌ها صحبت خواهیم کرد. در انتها نیز به بررسی رابط کاربری این پروژه می‌پردازیم.

* 1. آماده‌سازی سیستم

برای اجرای خزنده بر روی هر سیستم عامل، نیازمند نصب پایتون و کتاب‌خانه‌های استفاده‌شده در پروژه و پایگاه داده‌ی MySQL هستیم. با فعال‌سازی environment اجرا، پکیج‌ها را نصب و آماده‌سازی کنید.

* 1. اجرای خزنده‌ی کدها

برای اجرای خزنده‌ی کدها می توانید از دستورالعمل خزنده‌ی قبلی که به اختصار در اینجا آورده می‌شود استفاده کنید.

ابتدا پروژه را از GitLab دانلود کرده و با استفاده از فایل requirements پکیج‌های مورد نیاز را نصب کنید.

با اجرای دستور

Scrapy Crawl InstrumentCode

فایل instrumentCode.csv ساخته‌شده را در کنار فایل manage.py در پروژه‌ی اصلی کپی کنید. با ساختن یک پایگاه داده به نامی که در settings.py مشخص‌شده، می‌توانید به اجرای توابع پروژه بپردازید.

اجرای توابع اصلی

خزنده با استفاده از هر گونه shell در پایتون قابل اجرا است. پس از جای‌گذاری فایل csv ابتدا در ترمینال به دایرکتوری اصلی پروژه وارد شوید. فایل manage.py در این دایرکتوری قرار دارد. با استفاده از دستور زیر می‌توان shell را باز کرد:

Python manage.py shell

سپس باید سرویس‌های پروژه را با استفاده از دستورهای زیر ایمپورت کنیم:

From bourse.services.instrument\_services import\*

From bourse.services.others import\*

برای اجرای تابع crawl نیازمند دو تاریخ به فرمت استاندارد هستیم. پس با دستور

From datetime import date

تابع تولید تاریخ را از کتاب‌خانه‌ی مربوطه ایمپورت می‌کنیم؛ سپس می‌توانیم دو تاریخ برای شروع و پایان (اختیاری) کراول وارد کنیم. فرمت استاندارد تاریخ به صورت سال، ماه، روز و بدون صفر قبل اعداد تک‌رقمی است، به عنوان مثال برای شروع و پایان آرگومان‌های زیر را می‌توان وارد کرد:

Start= date(2020, 1, 1)

End= date(2020, 2, 2)

سپس با استفاده از تابع اصلی کراول، تمامی توابع اجرا می‌شوند:

Crawl(Start, End)

در صورت نیاز می‌توان برای موازی‌سازی استخراج اطلاعات، توابع را به صورت جداگانه نیز فراخوانی کرد که در این صورت باید هر روز دستور فراخوانی تابع را با آرگومان آن روز اجرا کنیم. به عنوان مثال:

Get\_day\_Price\_Threshold(Start)

یا

Get\_day\_Price\_Threshold(date(2020, 1, 1))

رابط گرافیکی

رابط گرافیکی مورد نیاز برای نظارت بر پروژه شامل یک صفحه‌ی پیاده‌سازی‌شده توسط react است که با استفاده از APIهای PHP از دیتابیس کار می‌کند. این وب‌سایت هم‌اکنون از طریق آی‌پی 185.126.200.119 قابل دسترسی است. این صفحه به سه بخش Header، Sidebar و Content تقسیم شده است.



پیاده‌سازی رابط گرافیکی

* + 1. هدر

هدر شامل یک ماژول جستجو برای یافتن نماد‌ها است که با نوشتن در آن، لیستی از نماد‌هایی که نام فارسی اختصاری آن‌ها شامل آن متن هستند را به نمایش می‌گذارد. با انتخاب هر کدام از این نمادها اطلاعات آن به نمایش در خواهد آمد. لازم به ذکر است از آن‌جا که در پایگاه داده‌ی شرکت مدیریت فناوری بورس تهران، نام اختصاری نمادها با حروف عربی ذخیره‌سازی شده است، یک نرمالایزر برای تبدیل حروف فارسی به عربی نیز در این ماژول به‌کار گرفته‌ شده‌است.

* + 1. منوی کناری

این منو شامل اطلاعات اولیه‌ی نماد، میانگین ده‌روزه‌ی خرید حقیقی و حقوقی و هم‌چنین حجم معاملات است.

* + 1. محتوای اصلی

این بخش شامل نمودار تغییر قیمت و اطلاعات خرید و فروش روزانه‌ی نماد در یک قسمت و در قسمت پایین نمودارهای مقایسه‌ای سرانه‌ی معاملات، حجم خرید و فروش و درصد سهم خرید و فروش برای سرمایه‌گذاران حقیقی و حقوقی است. نمودار تغییر قیمت اطلاعات کامل تهیه‌شده را به شکل شمعی که استاندارد نمودارهای مالی است به نمایش می‌گذارد. توسط دکمه‌ی تعدیل در زیر این نمودار می‌توان اطلاعات قیمت تعدیل‌شده را نیز مشاهده نمود.

خلاصه و جمع‌بندی

در این فصل به توضیح فرایند آماده‌سازی محیط اجرای خزنده‌ها و دستورات لازم برای اجرای هر دو خزنده پرداخته‌شد. هم‌چنین به شرح ویژگی‌های رابط کاربری نیز پرداختیم.

1. جمع‌بندی، نتيجه‌گيری و پيشنهادها

در این فصل نتیجه‌گیری‌های کلی حاصل‌شده در این تحقیق بیان می‌شود و محدودیت‌ها مورد بحث قرار میگیرد. هم‌چنین پیشنهادهایی برای ادامه‌ی مسیر به علاقمندان این حوزه ارائه خواهد شد.

* 1. جمع‌بندی

در اين تحقيق در گام نخست ما به مطالعه‌ی اولیه روی مفاهیم اولیه‌ی بازار بورس پرداختیم و با ساختار صفحات و معماری اطلاعات وب‌سایت مدیریت فناوری بورس تهران آشنا شدیم. سپس با مطالعه‌ درباره‌ی نحوه‌ی پیاده‌سازی خزنده‌ی قبلی متوجه اصلاحات ساختاری این وب‌سایت شدیم. بدین ترتیب با یافتن ایرادات خزنده‌ی قبلی، به این نتیجه رسیدیم که نیازمند اصلاحات ساختاری هستیم و در نتیجه، خزنده‌ی جدیدی را پیاده‌سازی نمودیم. در بازطراحی مدل داده‌ای و پایگاه داده سعی بر آن کردیم که حتی‌الامکان ساختار مدل داده‌ای قبل را حفظ کنیم. پس از پایان این فرایند با راه‌اندازی سرور و استخراج منظم داده‌ها از صحت این اطلاعات، اطمینان حاصل کردیم و سپس رابط کاربری ساده و مناسبی پیاده‌سازی و مستقر کردیم.

دستاوردها

ایجاد یک مدل داده‌ای کامل و به روز با قابلیت تغییرات گسترده از جمله حذف یا اضافه‌کردن اطلاعات دل‌خواه(با توجه به APIهایی که در اختیار داریم) مهم‌ترین دستاورد این پروژه است. از دیگر دستاوردهای این پروژه می‌توان به رابط گرافیکی ساده و مناسب برای بررسی صحت داده‌ها و هم‌چنین استفاده‌های آتی در تحلیل‌های مختلف اشاره‌کرد.

پیشنهادها

با توجه به تغییرات صورت‌گرفته، در ادامه‌ی روند این پروژه می‌توان اطلاعات دقیق‌تر و کامل‌تری از تمامی تغییرات و ویژگی‌های نمادها و سهام به دست آورد. اجرای منظم این خزنده و افزودن اطلاعات تمام معاملات می‌تواند در دسترسی به داده‌هایی دقیق‌تر و مفیدتر کمک شایانی کند. با استفاده از جداول کمکی مانند جدول میانگین‌های ده‌روزه و با گسترش این جدوال می‌توان داده‌های آماده برای اجرای روش‌های مختلف مبتنی بر هوش مصنوعی و یادگیری ماشین فراهم نمود. این روش‌ها در کنار روش‌های تحلیل بنیادین و تکنیکال می‌توانند محاسبات بهتری بر روی کاهش و افزایش ارزش سهام و در نتیجه راهنمای مناسبی برای مبادله‌ی هوشمند ارائه دهند.

1. مراجع

مراجع

[1] S. Madge, S. Bhatt, "Predicting stock price direction using support vector machines," Independent work report spring, 2015.

[2]Rubio D. (2017) Django Models. In: Beginning Django. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-2787-9\_7

[3] Yatskov, A.K., Varlamov, M.I. & Turdakov, D.Y. Extraction of Data from Mass Media Web Sites. Program Comput Soft 44, 344–352 (2018). https://doi.org/10.1134/S0361768818050092

[4] Hetland M.L. (2017) Python and the Web. In: Beginning Python. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-0028-5\_15

|  |  |  |
| --- | --- | --- |
| Optimized_image_813b5ec2.png | University of Tehran | Images__Logo_FE.gif |
| College of Engineering  School of Electrical and Computer Engineering  **Preparing and Launching the Tehran Stock Exchange Data Crawling and Processing System** | | |
| A thesis submitted to the Undergraduate Studies Office  In partial fulfillment of the requirements for  The degree of Master in  Computer Engineering | | |
| **By:**  **Ali Ghane**  **Supervisor:**  **Hesham Faili PhD** | | |

1. Tsetmc.com [↑](#footnote-ref-1)
2. Crawl [↑](#footnote-ref-2)
3. http://www.tsetmc.com/Loader.aspx?ParTree=111C1417 [↑](#footnote-ref-3)
4. Application Programming Interface [↑](#footnote-ref-4)
5. JavaScript Object Notation [↑](#footnote-ref-5)
6. Attribute-Value [↑](#footnote-ref-6)
7. Object-Relational-Mapping [↑](#footnote-ref-7)