|  |  |  |
| --- | --- | --- |
|  | دانشگاه تهران  پردیس دانشکده­های فنی  دانشکده مهندسی برق و کامپیوتر |  |
|  | | |
|  | | |
| سامانه پویش و آماده‌سازی داده‌های بورس تهران  پايان‌نامه براي دريافت درجه کارشناسی  در رشته مهندسی کامپیوتر گرايش فناوری اطلاعات | | |
| **نام:**  **مرضیه باقری‌نیا امیری**  **شماره دانشجویی:**  810197682 | | |
| **استاد راهنما:**  جناب آقای دکتر اسدپور | | |
|  | | |
| **شهریورماه 1401** | | |
|  | | |



|  |
| --- |
| **تعهدنامه اصالت اثر** |
| **باسمه تعالي** |
| **اينجانب مرضیه باقری‌نیا امیری تائيد مي كنم كه مطالب مندرج در اين پایان نامه حاصل تلاش 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 :**  **مرضیه باقری‌نیا امیری**  **امضاي دانشجو :** |

**چکيده**[[1]](#footnote-1)

بازارهای مالی در سراسر جهان از جذاب‌ترین و پرچالش‌ترین بسترها برای فعالیت‌های اقتصادی است؛ در ایران نیز بازار بورس از این قاعد مستثنی نبوده و با افزایش دانش عمومی نسبت به آن، رونق بیشتری گرفته است و افراد حقیقی و حقوقی پیوسته به دنبال راه‌حل‌هایی برای پیش‌بینی قیمت سهم‌ها و افزایش بازدهی و کسب سود از این بازار هستند. اولین چیزی که در همه روش‌های فنی به آن نیاز است یک مجموعه داده تمیز از اطلاعات مربوط به سهم‌های موجود در این بازار است؛ اطلاعات مربوط به سهم‌ها و سایر اطلاعات مربوط به بازار به صورت داده‌ی خام از طریق وبسایت [[2]](#footnote-2)سازمان بورس و اوراق بهادار تهران قابل دسترسی است. پراکندگی و کلیت داده‌ها در این وبسایت باعث شده است تا به یک سامانه پویش و پرادزش داده نیاز داشته باشیم که اطلاعات مورد نیاز روش‌ تحلیلی انتخابی ما برای پیش‌بینی قیمت سهم را از وبسایت مربوط پویش کرده و آن را پس از مرتب‌سازی، آماده پردازش نماید. هدف این پروژه این است که سامانه‌ای با ویژگی‌های مذکور را طراحی و آماده نماییم تا از خروجی آن بتوان در پروژه‌های آتی مربوز به تحلیل قیمتی سهام استفاده نمود.

**کلمات کلیدی: بورس، قیمت سهم، داده، مدل داده، خزنده**

**فهرست مطالب**

[فصل 1: مقدمه و بيان مساله 1](#_Toc112933966)

[1-1- مقدمه 2](#_Toc112933967)

[1-2- تاريخچه‌ای از موضوع تحقيق 2](#_Toc112933968)

[1-3- شرح مسئله تحقيق 2](#_Toc112933969)

[1-4- تعريف موضوع تحقيق 3](#_Toc112933970)

[1-5- اهداف و آرمان‌های کلی تحقيق 3](#_Toc112933971)

[1-6- روش انجام تحقیق 3](#_Toc112933972)

[1-7- ساختار پايان‌نامه 4](#_Toc112933973)

[فصل 2: مفاهيم اوليه و پیش‌نیازها 5](#_Toc112933974)

[2-1- مقدمه 6](#_Toc112933975)

[2-2- بررسی ساختار سایت شرکت مدیریت فناوری بورس تهران 6](#_Toc112933976)

[2-2-1- صفحه اصلی وبسایت 6](#_Toc112933977)

[2-2-2- صفحه مربوط به شاخص‌ها 6](#_Toc112933978)

[2-2-3- لیست کلیه نمادها 8](#_Toc112933979)

[2-2-4- صفحه مربوط به تاریخچه هر نماد 10](#_Toc112933980)

[2-3- ساختار کلی خزنده 10](#_Toc112933981)

[2-3-1 دریافت اطلاعات کلی نماد 10](#_Toc112933982)

[2-3-2 دریافت اطلاعات قیمتی معاملات یک نماد در یک تاریخ خاص 11](#_Toc112933983)

[2-3-3- دریافت اطلاعات معاملات سهامداران حقیقی و حقوقی یک نماد در یک تاریخ خاص 12](#_Toc112933984)

[2-3-4- دریافت اطلاعات شاخص‌ها 13](#_Toc112933985)

[2-4- خلاصه و جمع بندی 14](#_Toc112933986)

[فصل 3: ساختار، معماری اجزا و روش کارکرد خزنده 15](#_Toc112933987)

[3-1- مقدمه 16](#_Toc112933988)

[3-2- پایگاه داده 16](#_Toc112933989)

[3-2-1- جدول Instrument 16](#_Toc112933990)

[3-2-2- جدول Indexes 17](#_Toc112933991)

[3-2-3- جدول Codal 18](#_Toc112933992)

[3-2-4- جدول ClientTradeInfo 19](#_Toc112933993)

[3-2-5- جدول DayTradeSummary 20](#_Toc112933994)

[3-2-6- جدول ChangeCapital 20](#_Toc112933995)

[3-2-7- جدول Export 21](#_Toc112933996)

[3-3- ماژول خزنده 21](#_Toc112933997)

[3-3-1- تابع get\_instruments 21](#_Toc112933998)

[3-3-2- تابع get\_indexes 22](#_Toc112933999)

[3-3-3- تابع get\_day\_trade\_summaries 22](#_Toc112934000)

[3-3-4- تابع get\_client\_trade\_infos 22](#_Toc112934001)

[3-3-5- تابع get\_change\_capitals 22](#_Toc112934002)

[3-3-6- تابع calculate\_balanced\_price 23](#_Toc112934003)

[3-3-7- تابع create\_daily\_data 23](#_Toc112934004)

[3-3-8- تابع unique\_crawl 23](#_Toc112934005)

[3-3-9- تابع setup 23](#_Toc112934006)

[3-3-10- تابع crawl 23](#_Toc112934007)

[3-3-11- تابع add\_new\_instrument 24](#_Toc112934008)

[3-4- خلاصه و جمع‌بندی 24](#_Toc112934009)

[فصل 4: راه‌اندازی و اجرای خزنده 25](#_Toc112934010)

[4-1- مقدمه 26](#_Toc112934011)

[4-2- راه‌اندازی خزنده بر روی Localhost 26](#_Toc112934012)

[4-2-1- آماده‌سازی فایل‌های Setup اولیه 26](#_Toc112934013)

[4-2-1-1- تکمیل فایل indexes.csv 26](#_Toc112934014)

[4-2-1-2- تکمیل فایل ins.csv 28](#_Toc112934015)

[4-2-2- ساخت پایگاه داده مربوط به پروژه در mySQL 30](#_Toc112934016)

[4-2-3- اجرای خزنده 31](#_Toc112934017)

[4-2-4- اجرای سامانه پویش داده بر روی localhost 32](#_Toc112934018)

[4-3- کار با خزنده بر روی سرور Linux 34](#_Toc112934019)

[4-4- کار با سامانه پویش و آماده‌سازی داده‌های بورس تهران 35](#_Toc112934020)

[4-5- خلاصه و جمع‌بندی 36](#_Toc112934021)

[فصل 5: جمع‌بندی، نتيجه‌گيری و پيشنهادها 37](#_Toc112934022)

[5-1- جمع‌بندی 38](#_Toc112934023)

[5-2- نتيجه‌گيری 38](#_Toc112934024)

[5-2-1- دستاوردها 38](#_Toc112934025)

[5-2-2- پيشنهادها 38](#_Toc112934026)

[فصل 6: مراجع 39](#_Toc112934027)

فهرست شکل­ها

[شکل 1-2) صفحه اصلی وبسایت شرکت مدیریت فناوری بورس تهران 6](#_Toc112938358)

[شکل 2-2) انتخاب شاخص‌های منتخب 7](#_Toc112938359)

[شکل 3-2) لیست تمامی شاخص‌ها 7](#_Toc112938360)

[شکل 4-2) صفحه مربوط به شاخص‌ کل و کد یکتای آن 8](#_Toc112938361)

[شکل 5-2) لیست تمامی نمادهای بازار بورس ایران 9](#_Toc112938362)

[شکل 6-2) صفحه اصلی مربوط به نماد شبندر 9](#_Toc112938363)

[شکل 7-2) صفحه مربوط به تاریخچه سهم شبندر در تاریخ 06/08/2022 10](#_Toc112938364)

[شکل 8-2) تصویر خروجی API مربوط به اطلاعات کلی نماد شبندر در تاریخ 06/08/2022 11](#_Toc112938365)

[شکل 9-2) تصویر خروجی API مربوط به اطلاعات معاملات نماد شبندر در تاریخ 06/08/2022 12](#_Toc112938366)

[شکل 10-2) تصویر خروجی API مربوط به اطلاعات معاملات سهامداران نماد شبندر در تاریخ 06/08/2022 13](#_Toc112938367)

[شکل 11-2) تصویر خروجی API مربوط به اطلاعات شاخص کل 14](#_Toc112938368)

[شکل 12-3) مدل داده‌ای پایگاه داده 17](#_Toc112938369)

[شکل 13-4) انتخاب شاخص‌های منتخب 28](#_Toc112938370)

[شکل 14-4) لیست تمامی شاخص‌ها 28](#_Toc112938371)

[شکل 15-4) صفحه مربوط به شاخص‌ کل و کد یکتای آن 28](#_Toc112938372)

[شکل 16-4) تصویر فایل indexes.csv 29](#_Toc112938373)

[شکل 17-4) صفحه اصلی مربوط به نماد 30](#_Toc112938374)

[شکل 18-4) صفحه مربوط به سابقه معاملاتی نماد 30](#_Toc112938375)

[شکل 19-4) تصویر فایل نهایی ins.sv 31](#_Toc112938376)

[شکل 20-4) تنظیمات پایگاه داده در settings.py 32](#_Toc112938377)

[شکل 21-4) تصویر Shell مربوط به Django 33](#_Toc112938378)

[شکل 22-4) خروجی اجرای دستور runserver 34](#_Toc112938379)

[شکل 23-4) سامانه پویش و پردازش داده بورس تهران 34](#_Toc112938380)

[شکل 24-4) ترمینال سرور لینوکسی 35](#_Toc112938381)

[شکل 25-4) تصویر وبسایت سامانه پویش و آماده‌سازی داده‌های بورس تهران 36](#_Toc112938382)

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 تمامی سرمایه‌گذاران سعی دارند تا کارامدترین روش‌های تحلیل قیمتی سهام را برگزینند و از آن بهره ببرند؛ در تمامی این روش‌های تحلیلی، داده‌های مربوط به سهام رکن اصلی را تشکیل می‌دهند. هرچه این اطلاعات دقیق‌تر، جامع‌تر و دسته‌بندی‌شده باشد، تحلیل انجام‌شده بر روی آن نیز معتبرتر و قابل اتکا خواهد بود. بنابراین وجود یک سامانه که بتواند تمامی اطلاعات مربوط به یک سهم را به صورت مرتب، دسته‌بندی شده و قابل تحلیل در اختیار بگذارد، می‌تواند کمک شایانی به تحلیلگران و سرمایه‌گذاران بازار سهام بکند.

* 1. تاريخچه‌ای از موضوع تحقيق

این پروژه در امتداد و تکمیل پروژه‌ی «راه‌اندازی و آماده‌سازی سامانه‌ی پویش و پردازش داده‌های بورس تهران» که توسط آقای علی قانع زیر نظر دکتر هشام فیلی در شهریورماه 1400 در دانشکده‌ی برق و کامپیوتر دانشگاه تهران ارائه شده، نوشته شده‌است.

* 1. شرح مسئله تحقيق

فلسفه کلیدی شکل‌گیری بورس در جهان، ایجاد بازاری منسجم، قانونمند، پیوسته و شفاف برای تجمیع سرمایه‌های خرد و کلان سرمایه‌گذاران به‌منظور تأمین مالی شرکت‌ها و پروژه‌های آن‌ها است. در حقیقت مبنای اساسی تشکیل بازار بورس، همان مفهوم اولیه شراکت و تقسیم سود است. به این معنی که اگر فردی سهام شرکتی را خریداری کند، به نسبت تعداد سهامی که در اختیار دارد، مالک شرکت شده و ضمن بهره‌مندی از سودهای کسب‌شده شرکت، می‌تواند از نوسان قیمت سهام آن نیز منتفع شود. بنابراین، با توجه به آنچه گفته شد، سهامداران و سرمایه‌گذاران در بورس، باید بتوانند وضعیت سهم شرکت‌های مختلف را بررسی نموده و آینده قیمتی آن‌ها را با دقت قابل قبولی پیش‌بینی نمایند و در صورتی که سهمی را مستعد رشد دیدند آن را خریداری نمایند؛ پس می‌توان گفت که پیش‌بینی قیمت یک سهم یک مسئله بسیار با اهمیت تلقی می‌شود که تاکنون راه‌حل‌های متفاوتی نیز برای آن ارائه شده است.

با گذشت زمان و افزایش گردش اطلاعات، استفاده از داده‌های تاریخی مرتبط با سهم‌ها به‌منظور پیش‌بینی روند قیمتی آینده آن‌ها بسیار رایج شده است؛ بنابراین نیاز به یک سامانه تجمیع و آماده‌سازی اطلاعات مربوط به سهام به صورت یکپارچه و مرتب و در فرمت‌های قابل استفاده در سامانه‌های تحلیلی بیش از پیش به چشم می‌آید و باید مرتفع گردد.

* 1. تعريف موضوع تحقيق

در این پروژه، قصد داریم تا یک سامانه‌ای را توسعه دهیم که داده‌های بازار بورس تهران را در ساختاری مناسب جمع‌آوری نموده و و به منظور استفاده در سامانه‌های تحلیلی ارائه نماید. بدین منظور یک خزنده**[[3]](#footnote-3)** جهت جمع‌آوری داده‌های وبسایت سازمان بورس مورد نیاز است؛ مدل داده‌ای این خزنده، دقت داده‌‌های جمع‌آوری‌شده توسط آن، سرعت جمع‌آوری داده و کاربردی بودن آن می‌تواند کمک شایانی در کاربردپذیری این خزنده داشته باشد. در این رابطه پروژه سعی شده است تا با بررسی دقیق منبع خزش (وبسایت سازمان بورس) و نیز خزنده‌‌های قبلی، چالش‌ها و کاستی‌های موجود در خزنده‌های قبلی رفع شده و سامانه‌ای طراحی شود که اطلاعات را با سرعت و دقت از وبسایت سازمان بورس جمع‌آوری کرده، اطلاعات مفید را از درون آن استخراج نماید و در قالبی قابل تحلیل و مرتب برای بارگیری و استفاده در برنامه‌های دیگر آماده کرده و بر روی بستر یک وبسایت با رابط کاربری مناسب نمایش دهد.

* 1. اهداف و آرمان‌های کلی تحقيق

اهداف کلی این پروژه شامل رفع ایرادات و ایجاد تغییرات ساختاری مورد نیاز بر روی خزنده‌های موجود قبلی و توسعه و استقرار سامانه‌ای با رابط کاربری مناسب برای دریافت اطلاعات تولیدی توسط خزنده است.

* 1. روش انجام تحقیق

روش انجام این پروژه در گام‌های زیر خلاصه می‌شود:

* بررسی دقیق وبسایت بازار بورس ایران و راه‌های واکشی اطلاعات از آن
* بررسی و یافتن ایرادات و نواقص خزنده‌ی قبلی
* بازطراحی و پیاده‌سازی خزنده
* طراحی و پیاده‌سازی رابط گرافیکی و استقرار آن
* راه‌اندازی سرور و استخراج منظم داده‌ها
  1. ساختار پايان‌نامه

فصل دوم، شامل بررسی دقیق سایت سازمان بورس و اطلاعات موجود در آن، تشریح اجزای اصلی خزنده و مدل داده و مروری بر پيش‌زمينه‌های مورد نياز برای درک هرچه بهتر ساختار پروژه است.

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

فصل چهارم، شامل دستورالعمل راه‌اندازی و اجرای مجدد پروژه بر روی Localhost و نیز روش کار با سروری که سامانه بر روی آن راه‌اندازی شده است، می‌باشد. هم‌چنین در این فصل به توضیحاتی در مورد رابط کاربری و نحوه استفاده از سامانه نیز خواهیم پرداخت.

فصل پنجم، شامل نتيجه‌گيری‌های کلی حاصل‌شده از اين پروژه و نیز پيشنهادهایی برای ادامه‌ی مسير به علاقمندان اين حوزه می‌باشد.

1. مفاهيم اوليه و پیش‌نیازها

در این فصل، مقدمات، مفاهيم اوليه و پيش‌زمينه‌هايی که جهت درک بهتر توضیحات آتی و نیز سهولت کار با سامانه به آن نیاز خواهد بود، اين پایان‌نامه مورد نياز است، ارائه خواهد شد.

مقدمه

*همانطور که پیشتر بدان اشاره شد، در این پروژه قصد داریم تا خزنده‌ای را توسعه دهیم که اطلاعات را از وبسایت سازمان بورس ایران جمع‌آوری و جهت بارگیری آماده می‌نماید؛ بنابراین برای درک بهتر از عملکرد و ساختار خزنده لازم است تا با ساختار این وبسایت و نحوه چینش داده‌ها در آن بیشتر آشنا شویم.*

* 1. بررسی ساختار سایت شرکت مدیریت فناوری بورس تهران
     1. صفحه اصلی وبسایت[[4]](#footnote-4)

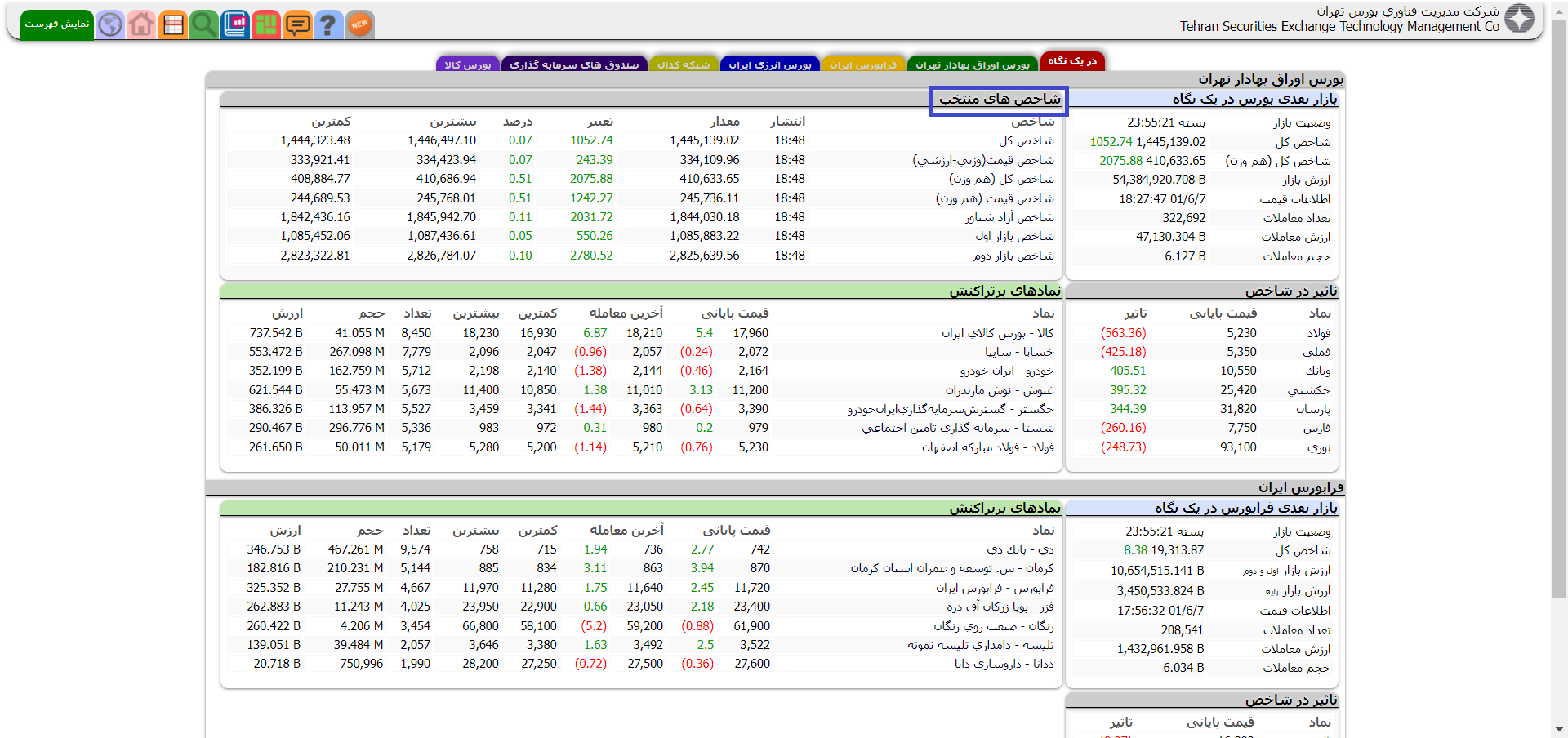
این صفحه که با این [لینک](http://www.tsetmc.com/Loader.aspx?ParTree=15) قابل دسترسی است، شامل خلاصه اطلاعات بازار سهام (بورس و فرابورس) اعم از مقدار شاخص‌ها، ارزش بازار، تعداد معاملات، حجم معاملات، لسیت نمادهای با بیشترین تاثیر در مقدار شاخص، لیست نمادهای پرتراکنش و ... در آخرین روز معاملاتی می‌باشد. تصویر این صفحه در ادامه در شکل (2-1) قابل مشاهده است:



شکل 1-2) صفحه اصلی وبسایت شرکت مدیریت فناوری بورس تهران

* + 1. صفحه مربوط به شاخص‌ها

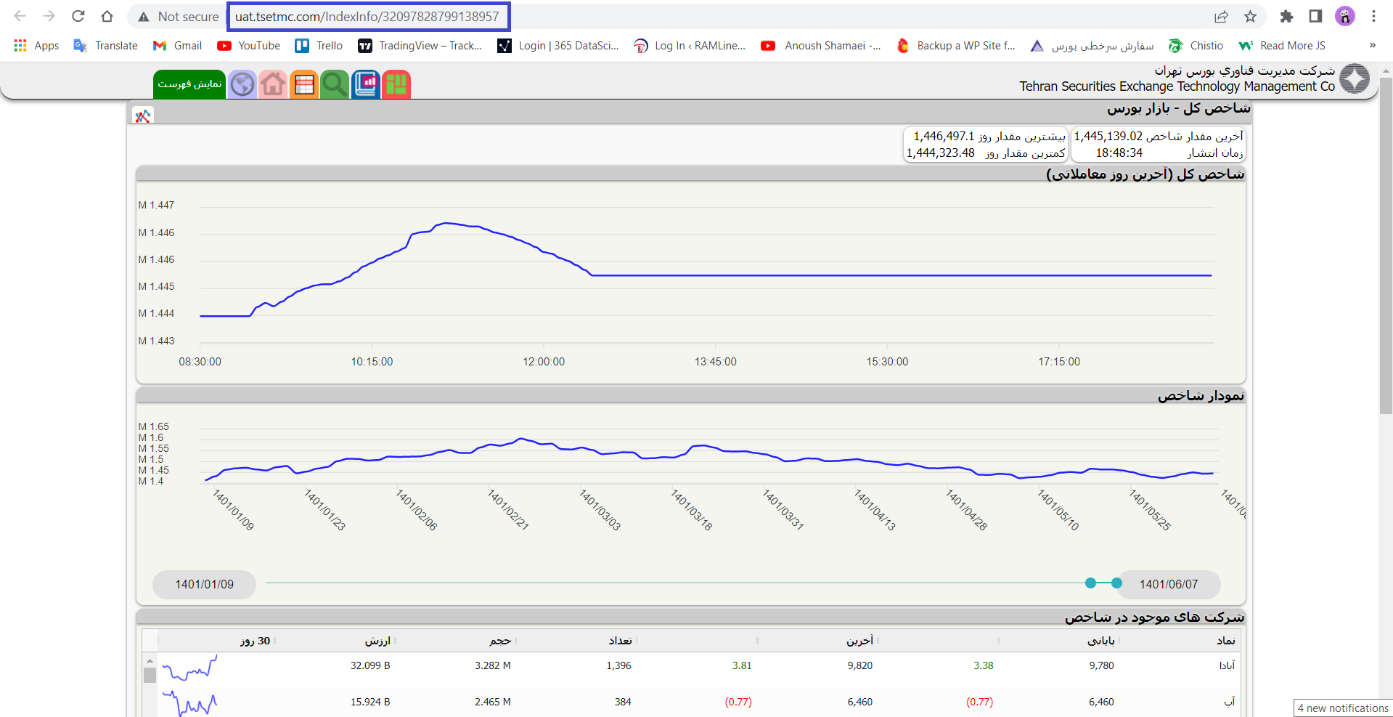
با کلیک بر روی عبارت «شاخص های منتخب» در صفحه اصلی وبسایت (شکل 2-2)، لیست تمامی شاخص‌های موجود در بازار اعم از شاخص کل، شاخص هموزن، شاخص‌های مربوط به صنایع مختلف و ... نمایان می‌شود (شکل 3-2) که با کلیک بر روی هر کدام وارد صفحه مربوط به آن شاخص می‌شویم؛ در انتهای url مربوط به صفحه هر شاخص کد یکتای آن شاخص قرار دارد که این کد در ادامه برای Crawl کردن تاریخچه اطلاعات این شاخص مورد نیاز است. در ادامه در (شکل 4-2) تصویر صفحه مربوط به شاخص کل بورس قابل مشاهده است:



شکل 2-2) انتخاب شاخص‌های منتخب



شکل 3-2) لیست تمامی شاخص‌ها



شکل 4-2) صفحه مربوط به شاخص‌ کل و کد یکتای آن

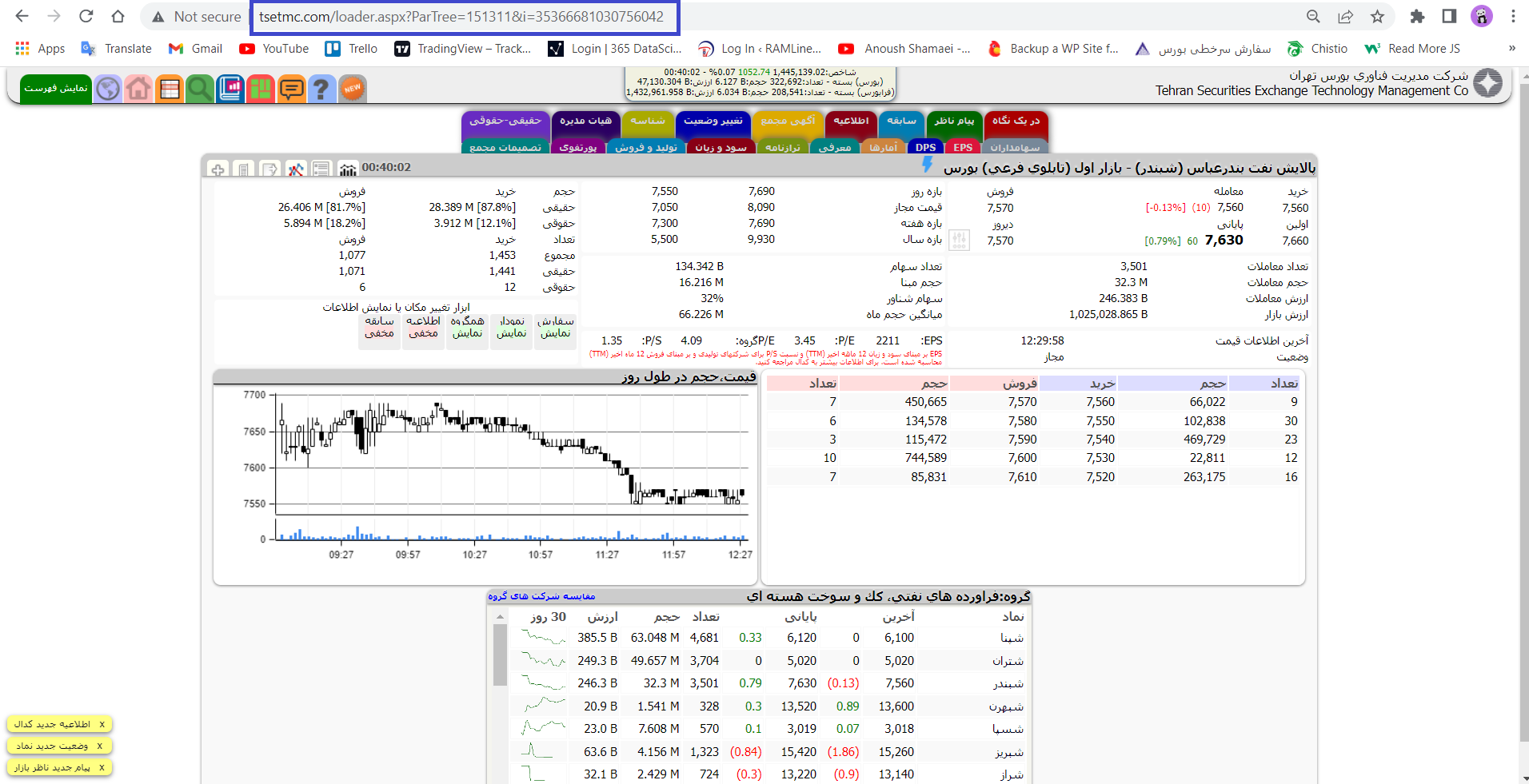
همانطور که در (شکل 4-2) قابل مشاهده است، url مربوط به صفحه شاخص کل <http://uat.tsetmc.com/IndexInfo/32097828799138957> می‎باشد که عدد انتهایی آن یعنی 32097828799138957 کد یکتای متتاظر با شاخص کل است.

* + 1. لیست کلیه نمادها[[5]](#footnote-5)

این صفحه که با این لینک قابل دسترسی است، شامل فهرست تمامی نمادهای موجود در بازار بورس ایران است (شکل 5-2) که با کلیک بر روی ستون نام یا نماد هر سهم وارد صفحه مربوط به آن سهم می‌شویم؛ در انتهای url مربوط به صفحه هر سهم کد یکتای آن سهم قرار دارد که این کد در ادامه برای Crawl کردن تاریخچه اطلاعات این سهم مورد نیاز است. در ادامه در (شکل 6-2) تصویر صفحه مربوط به نماد شبندر قابل مشاهده است:



شکل 5-2) لیست تمامی نمادهای بازار بورس ایران



شکل 6-2) صفحه اصلی مربوط به نماد شبندر

همانطور که در (شکل 6-2) قابل مشاهده است، url مربوط به صفحه نماد شبندر <http://www.tsetmc.com/loader.aspx?ParTree=151311&i=35366681030756042> می‎باشد که عدد انتهایی آن یعنی 35366681030756042 کد یکتای متتاظر با این نماد است.

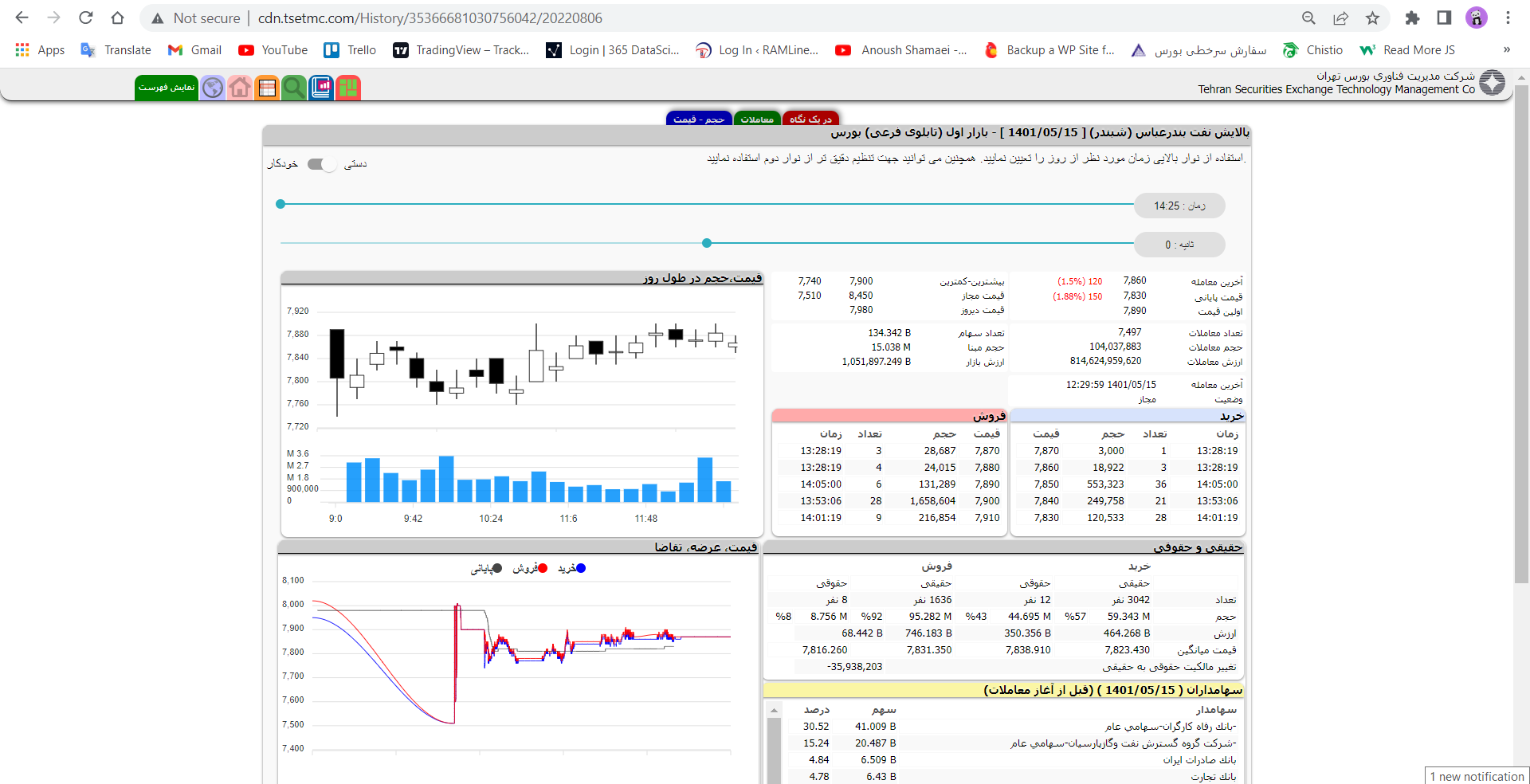
* + 1. صفحه مربوط به تاریخچه هر نماد

قالب URL مربوط به صفحه تاریخچه هر نماد به صورت زیر است:

<http://cdn.tsetmc.com/History/Instrument_Id/Date>

که در آن Instrument\_Id کد یکتای مربوط به نماد مورد نظر و Date تاریخی (به میلادی با فرمت yymmdd) است که می‌خواهیم اطلاعات نماد در آن روز را واکشی کنیم؛ برای مثال اطلاعات نماد شبندر در روز 2022/08/06 با URL زیر قابل استخراج است و تصویر آن نیز در ادامه در (شکل 7-2) قابل مشاهده است:

<https://cdn.tsetmc.com/History/35366681030756042/20220806>



شکل 7-2) صفحه مربوط به تاریخچه سهم شبندر در تاریخ 06/08/2022

* 1. ساختار کلی خزنده

با توجه به بررسی‌های صورت گرفته از وبسایت سازمان بورس، خزنده در چهار مرحله جداگانه و با استفاده از APIهای مختص هر مرحله تمامی اطلاعات موردنیاز سامانه را از وبسایت واکشی و در پایگاه داده ذخیره می‌نماید که در ادامه به شرح هر یک از این مراحل خواهیم پرداخت:

### 2-3-1 دریافت اطلاعات کلی نماد

در این مرحله لازم است تا نام سهم، نام نماد آن، کد نماد، حجم مبنای آن، سرمایه شرکت، نوع بازار (بورس - فرابورس) و ... ثبت گردد تا در مراحل بعدی که اطلاعات مربوط به معاملات انجام شده یک سهم در تاریخ‌های متفاوت ثبت می‌شود، بدان ارجاع گردد. بدین منظور از URL با فرمت زیر استفاده می‌نماییم که در آن Instrument\_Id کد یکتای مربوط به نماد مورد نظر و Date تاریخی (به میلادی با فرمت yymmdd) است که می‌خواهیم اطلاعات نماد در آن را واکشی کنیم (این تاریخ تنها در مقدار حجم مبنا و سرمایه شرکت تاثیرگذار است).

<http://cdn.tsetmc.com/api/Instrument/GetInstrumentHistory/Instrument_Id/Date>

برای مثال اطلاعات کلی نماد شبندر در روز 2022/08/06 با URL زیر قابل استخراج است و تصویر خروجی آن که در قالب JSON[[6]](#footnote-6) آماده شده است نیز در ادامه در (شکل 8-2) قابل مشاهده است:

<http://cdn.tsetmc.com/api/Instrument/GetInstrumentHistory/35366681030756042/20220806>



شکل 8-2) تصویر خروجی API مربوط به اطلاعات کلی نماد شبندر در تاریخ 06/08/2022

### 2-3-2 دریافت اطلاعات قیمتی معاملات یک نماد در یک تاریخ خاص

در این مرحله لازم است تا اطلاعات قیمتی مربوط به معاملات یک سهم در تاریخ‌های متفاوت را واکشی نماییم؛ بدین منظور از URL با فرمت زیر استفاده می‌نماییم که در آن Instrument\_Id کد یکتای مربوط به نماد مورد نظر و Date تاریخی (به میلادی با فرمت yymmdd) است که معاملات در آن انجام گرفته است.

<http://cdn.tsetmc.com/api/ClosingPrice/GetClosingPriceDaily/instrument_id/Date>

برای مثال اطلاعات معاملاتی نماد شبندر در روز 2022/08/06 با URL زیر قابل استخراج است و تصویر خروجی آن که در قالب JSON آماده شده است نیز در ادامه در (شکل 9-2) قابل مشاهده است:

<http://cdn.tsetmc.com/api/ClosingPrice/GetClosingPriceDaily/35366681030756042/20220806>



شکل 9-2) تصویر خروجی API مربوط به اطلاعات معاملات نماد شبندر در تاریخ 06/08/2022

### 2-3-3- دریافت اطلاعات معاملات سهامداران حقیقی و حقوقی یک نماد در یک تاریخ خاص

در این مرحله لازم است تا اطلاعات مربوط به معاملات سهامداران حقیقی و حقوقی یک نماد اعم از حجم، ارزش و تعداد معاملات مختص به هر گروه در تاریخ‌های متفاوت را واکشی نماییم؛ بدین منظور از URL با فرمت زیر استفاده می‌نماییم که در آن Instrument\_Id کد یکتای مربوط به نماد مورد نظر و Date تاریخی (به میلادی با فرمت yymmdd) است که معاملات در آن انجام گرفته است.

<http://cdn.tsetmc.com/api/ClientType/GetClientTypeHistory/Instrument_id/Date>

برای مثال اطلاعات معاملات سهامداران حقیقی و حقوقی نماد شبندر در روز 2022/08/06 با URL زیر قابل استخراج است و تصویر خروجی آن که در قالب JSON آماده شده است نیز در ادامه در (شکل 10-2) قابل مشاهده است:

<http://cdn.tsetmc.com/api/ClientType/GetClientTypeHistory/35366681030756042/20220806>



شکل 10-2) تصویر خروجی API مربوط به اطلاعات معاملات سهامداران نماد شبندر در تاریخ 06/08/2022

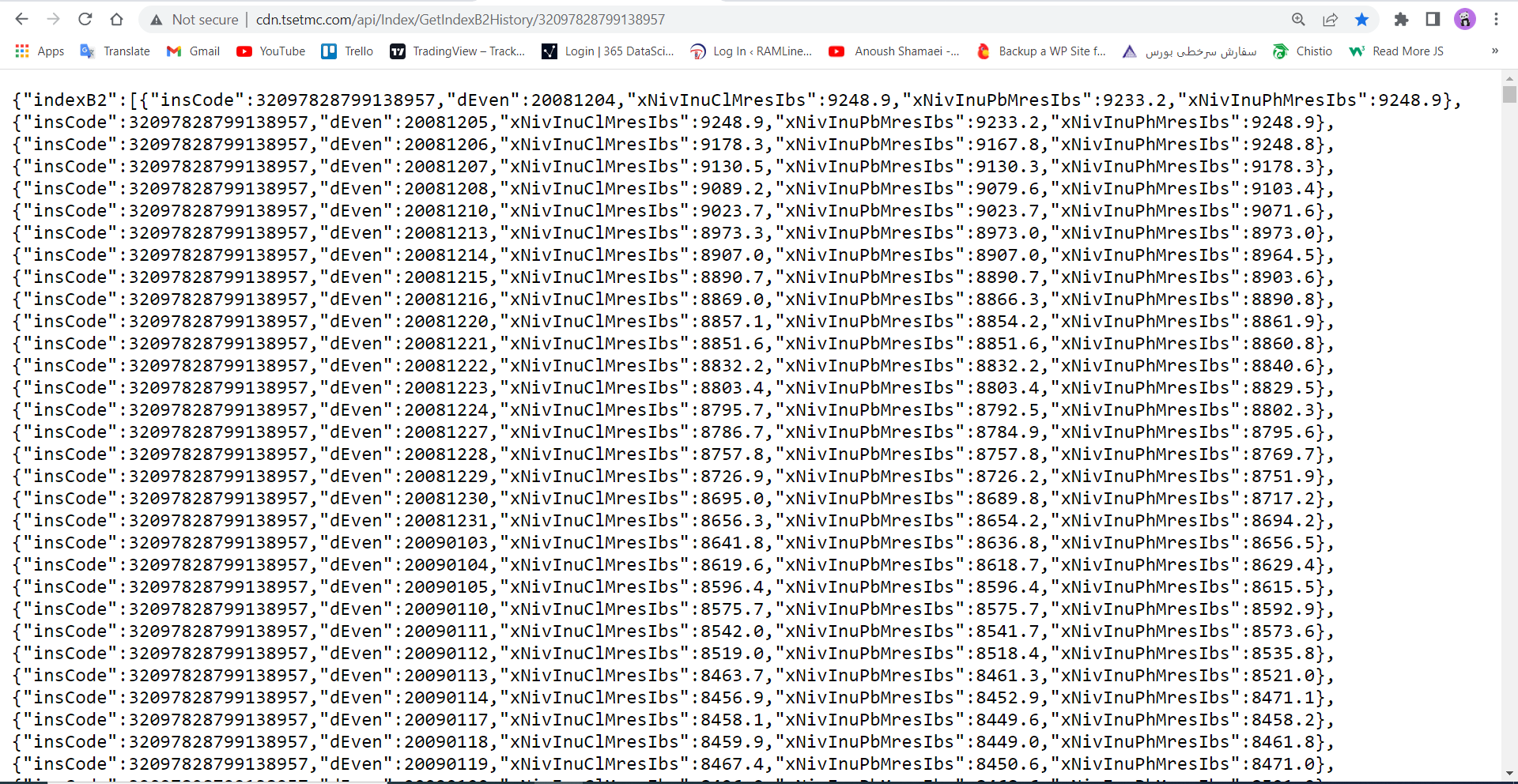
### 2-3-4- دریافت اطلاعات شاخص‌ها

در این مرحله لازم است تا اطلاعات مربوط به شاخص کل، شاخص هموزن و شاخص‌های صنایع مختلف را واکشی نماییم تا در ادامه آن‌ها را به اطلاعات هر نماد بیفزاییم؛ بدین منظور از URL با فرمت زیر استفاده می‌کنیم که در آن Index\_Id کد یکتای مربوط به شاخص مورد نظر است.

<http://cdn.tsetmc.com/api/Index/GetIndexB2History/Index_Id>

برای مثال اطلاعات مربوط به تاریخچه کلی شاخص کل با URL زیر قابل استخراج است و تصویر خروجی آن که در قالب JSON آماده شده است نیز در ادامه در (شکل 11-2) قابل مشاهده است:

<http://cdn.tsetmc.com/api/Index/GetIndexB2History/32097828799138957>



شکل 11-2) تصویر خروجی API مربوط به اطلاعات شاخص کل

* 1. خلاصه و جمع بندی

در اين فصل با ساختار کلی وبسایت سازمان بورس ایران و نحوه ذخیره‌سازی و چینش اطلاعات بازار در آن و نیز اجزای خزنده همخوان با ساختار وبسایت مذکور آشنا شدیم.

1. ساختار، معماری اجزا و روش کارکرد خزنده

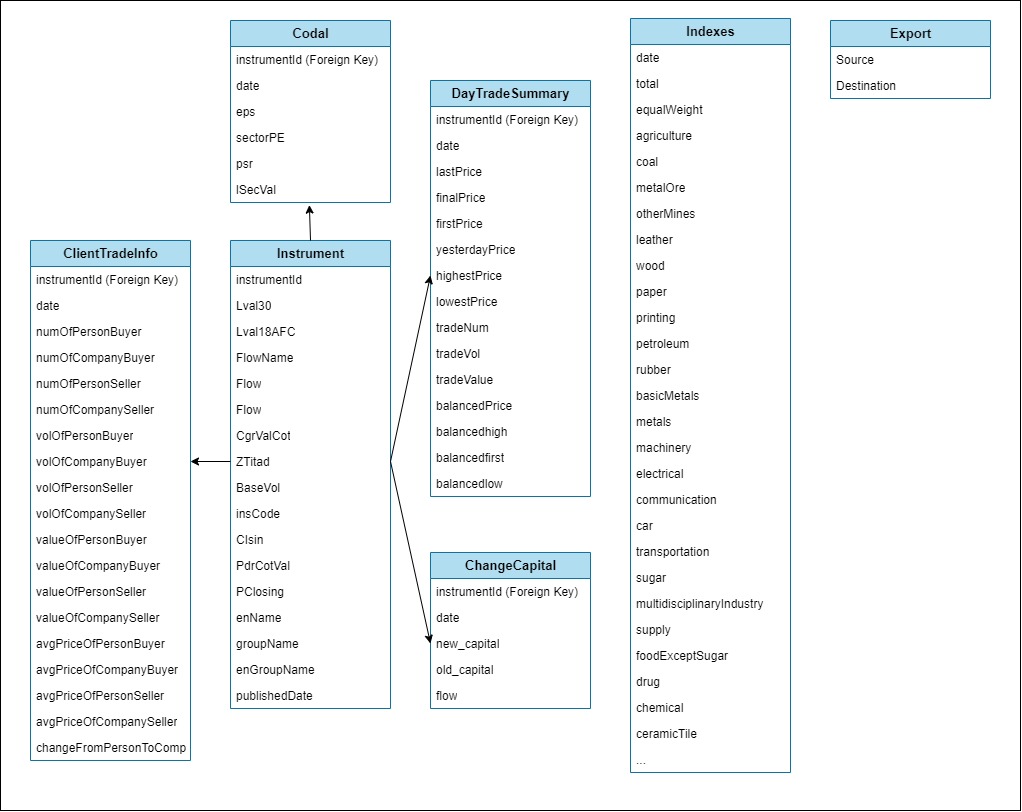
در این فصل، به تفصیل درباره ساختار و اجزای مختلف خزنده از طراحی تا پیاده‌سازی و اجرا توضیح داده خواهد شدد.

* 1. مقدمه

خزنده تحت فریم‌ورک جنگو[[7]](#footnote-7) طراحی و توسعه یافته است؛ جنگو بر اساس معماری MVT بنا نهاده شده است که مخفف عبارت مدل، نمایش و الگو است. هم‌چنین قابلیت ORM[[8]](#footnote-8) جنگو بستر را برای هماهنگ‌سازی با پایگاه داده‌ی MySQL (که در این پروژه از آن استفاده می‌کنیم) فراهم می‌آورد. در این فصل به تفصیل درباره قسمت‌های مختلف این خزنده از پایگاه داده تا ماژول‌های عملیاتی توضیح خواهیم داد.

* 1. پایگاه داده

پایگاه داده پیاده‌سازی شده در این پروژه از نوع MySQL است و شامل هشت جدول مجزا می‌باشد؛ نام این پایگاه داده tehran\_stock\_exchange می‌باشد و تنظیمات مربوط به آن (Username و Password) در فایل settings.py موجود در فولدر پروژه Django قرار داده شده است. در ادامه به ترتیب هر یک از جداول این پایگاه داده را معرفی و Fieldهای آن‌ها را تشریح خواهیم کرد. همچنین تصویر مدل داده‌ای این پایگاه داده در ادامه در (شکل 12-3) قابل مشاهده است.



شکل 12-3) مدل داده‌ای پایگاه داده

### 3-2-1- جدول Instrument

این جدول اطلاعات مربوط به هر نماد را ذخیره می‌کند؛ در واقع اغلب جداول پایگاه داده از طریق ارجاع و دسترسی به این جدول تکمیل می‌شوند. این جدول شامل 17 ویژگی[[9]](#footnote-9) می‌باشد که لیست این ویژگی‌ها و توضیح مربوط به هریک در ادامه آمده است:

instrumentId: کد یکتای مربوط به نماد

Lval30: نام فارسی شرکت

Lval18AFC: نام فارسی نماد

FlowName: نام بازاری که نماد در آن قرار دارد؛ مانند بازار اول (تابلوی اصلی)، بازار دوم بورس و ...

Flow: کد بازار

CgrValCot: کد گروه نماد

ZTitad: تعداد سهام

BaseVol: حجم مبنا

insCode: این کد برای هر نماد یکتا بوده و در هنگام اضافه شدن هر نماد جدید به بورس، در بانک اطلاعاتی وبسایت شرکت بورس ساخته می شود.

CIsin: کد دوازده رقمی شرکت

PClosing: آخرین قیمت

enName: نام انگلیسی نماد

groupName: نام گروه (صنعت) نماد

enGroupName: نام انگلیسی گروه (صنعت) نماد

publishedDate: تاریخ عرضه اولیه نماد

updated\_at: آخرین تاریخی که Object مربوط به نماد در پایگاه داده به روزرسانی شده است.

### 3-2-2- جدول Indexes

این جدول شامل اطلاعات تمامی شاخص‌ها در یک تاریخ خاص است؛ این جدول شامل 17 ویژگی می‌باشد که لیست این ویژگی‌ها و توضیح مربوط به هریک در ادامه آمده است:

date: تاریخی که مقدار شاخص‌ها در آن را ثبت کردیم.

Total: شاخص کل

equalWeight: شاخص هموزن

agriculture: شاخص زراعت و خدمات وابسته

coal: شاخص زغال‌سنگ

metalOre: شاخص استخراج کانه‌های فلزی

otherMines: شاخص استخراج سایرمعادن

leather: شاخص محصولات چرمی

Wood: شاخص محصولات چوبی

Paper: شاخص محصولات کاغذی

Printing: شاخص انتشار، چاپ و تکثیر

Petroleum: شاخص فراورده‌های نفتی

Rubber: شاخص لاستیک و پلاستیک

basicMetals: شاخص فلزات اساسی

metals: شاخص محصولات فلزی

Machinery: شاخص ماشین‌آلات و تجهیزات

Electrical: شاخص دستگاه‌های برقی

Communication: شاخص ساخت دستگاه‌های ارتباطی

Car: شاخص خودرو

Transportation: شاخص حمل‌ونقل

Sugar: شاخص قند و شکر

multidisciplinaryIndustry: شاخص شرکت‌های چندرشته‌ای صنعتی

supply: شاخص عرضه برق، گاز و آب

foodExceptSugar: محصولات غذایی به جز قند و شرکت

drug: شاخص محصولات دارویی

Chemical: شاخص محصولات شیمیایی

ceramicTile: شاخص کاشی و سرامیک

cement: شاخص سیمان

nonMetallicMineral: شاخص کانی‌های غیرفلزی

investment: شاخص سرمایه‌گذاری

bank: شاخص بانک‌ها

otherFinancial: شاخص سایر واسطه‌گری‌های مالی

radio: شاخص مخابرات

Financial: شاخص بیمه و صندوق‌های بازنشستگی به جز تامین اجتماعی

managingFinancialMarkets: شاخص فعالیت‌های کمکی به نهادهای واسط

bulkMaker: شاخص انبوه‌سازی

Computer: شاخص رایانه

It: شاخص اطلاعات و ارتباطات

Engineering: شاخص فنی و مهندسی

Insurance: شاخص بیمه

Retail: شاخص خرده‌فروشی

### 3-2-3- جدول Codal

این جدول شامل اطلاعات مربوط به کدال هر نماد در یک تاریخ خاص است؛ این جدول شامل 6 ویژگی می‌باشد که لیست این ویژگی‌ها و توضیح مربوط به هریک در ادامه آمده است:

Date: تاریخی که مقدار اطلاعات Codal نماد در آن را ثبت کردیم.

instrumentId: کد یکتای مربوط به نمادی که اطلاعات Codal آن را ثبت می‌کنیم.

Eps: مقدار EPS نماد

SectorPE: مقدار P/E نماد

Psr: مقدار P/S نماد

LSecVal: نام گروه (صنعت) نماد

### 3-2-4- جدول ClientTradeInfo

این جدول شامل اطلاعات معاملات مشتریان حقیقی و حقوقی یک نماد در یک روز معاملاتی خاص است؛ این جدول شامل 23 ویژگی می‌باشد که لیست این ویژگی‌ها و توضیح مربوط به هریک در ادامه آمده است:

Date: تاریخ روز معاملاتی

InstrumentId: Object مربوط به نماد (به صورت Foreign Key)

numOfPersonBuyer: تعداد خریداران حقیقی

numOfCompanyBuyer: تعداد خریداران حقوقی

numOfPersonSeller: تعداد فروشندگان حقیقی

numOfCompanySeller: تعداد فروشندگان حقوقی

volOfPersonBuyer: حجم معاملات خریداران حقیقی

volOfCompanyBuyer: حجم معاملات خریداران حقوقی

volOfPersonSeller: حجم معاملات فروشندگان حقیقی

volOfCompanySeller: حجم معاملات فروشندگان حقوقی

volPercentageOfPersonBuyer: درصد حجم معاملات خریداران حقیقی

volPercentageOfCompanyBuyer: درصد حجم معاملات خریداران حقوقی

volPercentageOfPersonSeller: درصد حجم معاملات فروشندگان حقیقی

volPercentageOfCompanySeller: درصد حجم معاملات فروشندگان حقوقی

valueOfPersonBuyer: ارزش معاملات خریداران حقیقی

valueOfCompanyBuyer: ارزش معاملات خریداران حقوقی

valueOfPersonSeller: ارزش معاملات فروشندگان حقیقی

valueOfCompanySeller: ارزش معاملات فروشندگان حقوقی

avgPriceOfPersonBuyer: قیمت میانگین خریداران حقیقی

avgPriceOfCompanyBuyer: قیمت میانگین خریداران حقوقی

avgPriceOfPersonSeller: قیمت میانگین فروشندگان حقیقی

avgPriceOfCompanySeller: قیمت میانگین فروشندگان حقوقی

changeFromPersonToCompany: درصد تغییر مالکیت حقیقی به حقوقی

### 3-2-5- جدول DayTradeSummary

این جدول شامل اطلاعات مربوط به معاملات یک نماد در یک روز معاملاتی خاص است؛ این جدول شامل 23 ویژگی می‌باشد که لیست این ویژگی‌ها و توضیح مربوط به هریک در ادامه آمده است:

instrumentId: Object مربوط به نماد (به صورت Foreign Key)

date: تاریخ روز معاملاتی

lastPrice: آخرین قیمت

finalPrice: قیمت پایانی

firstPrice: اولین قیمت

yesterdayPrice: قیمت پایانی روز معاملاتی قبل

highestPrice: بالاترین قیمت

lowestPrice: پایین‌ترین قیمت

tradeNum: تعداد معاملات

tradeVol: حجم معاملات

tradeValue: ارزش معاملات

balancedPrice: قیمت پایانی تعدیل‌شده

balancedhigh: بالاترین قیمت تعدیل‌شده

balancedfirst: اولین قیمت تعدیل‌شده

balancedlow: پایین‌ترین قیمت تعدیل‌شده

### 3-2-6- جدول ChangeCapital

این جدول شامل اطلاعات مربوط به افزایش سرمایه هر نماد است؛ این جدول شامل 5 ویژگی می‌باشد که لیست این ویژگی‌ها و توضیح مربوط به هریک در ادامه آمده است:

instrumentId: Object مربوط به نماد (به صورت Foreign Key)

date: تاریخ افزایش سرمایه

new\_capital: سرمایه جدید

old\_capital: سرمایه قبلی

flow: کد بازار نماد

### 3-2-7- جدول Export

این جدول برای ذخیره اطلاعات مربوط به فرم استفاده شده در view تعبیه شده است؛ این جدول شامل 2 ویژگی می‌باشد که لیست این ویژگی‌ها و توضیح مربوط به هریک در ادامه آمده است:

Source: تاریخ شروع دریافت اطلاعات نمادها

Destination: تاریخ پایان دریافت اطلاعات نمادها

* 1. ماژول خزنده

دریافت اطلاعات در خزنده از طریق APIهایی که سایت tsetmc تعبیه کرده است صورت می‌پذیرد و در توابع مختلفی که در ماژول مربوط به خزنده پیاده‌سازی شده است، با فراخوانی این APIها جداول پایگاه داده را به ترتیب تکمیل خواهیم کرد. کدهای مربوط به ماژول خزنده در فولدر bourse/services قابل دسترسی است؛ این فولدر حاوی سه فایل اصلی می‌باشد که عبارتند از: constants.py که شامل مقادیر ثابت (و بعضا پرتکرار) استفاده شده در توابع ماژول خزنده می‌باشد، functions.py که حاوی تمامی توابع فرعی استفاده شده در توابع اصلی ماژول خزنده می‌باشد (این کار به منظور افزایش خوانایی کدها انجام گرفته است) و crawlers.py که شامل تمامی توابعی است که بخش‌های مختلف crawl اطلاعات را انجام و پایگاه‌داده را پر می‌کنند. در ادامه به تشریح نحوه عملکرد توابع اصلی خزنده (توابع موجود در فایل crawlers.py می‌پردازیم):

* + 1. تابع get\_instruments

این تابع به منظور دریافت اطلاعات مربوط به نمادها تعبیه شده است؛ در این تابع، ابتدا کد یکتای مربوط به نمادها از فایل ins.csv خوانده می‌شود و سپس با استفاده از تابع کمکی add\_new\_ins و با دریافت اطلاعات از API با URL با فرمت [http://cdn.tsetmc.com/api/Instrument/GetInstrumentHistory/{Instrument\_Id}/{Date}](http://cdn.tsetmc.com/api/Instrument/GetInstrumentHistory/%7bInstrument_Id%7d/%7bDate%7d) یک Object از نوع Instrument می‌سازد؛ سپس با استفاده از تابع کمکی add\_new\_codal و نیز دریافت اطلاعات از API با URL با فرمت [http://www.tsetmc.com/loader.aspx?ParTree=151311&i={Instrument\_id](http://www.tsetmc.com/loader.aspx?ParTree=151311&i=%7bInstrument_id) یک Object از نوع Codal ساخته و اطلاعات بنیادی سهم مورد نظر را ثبت می‌کنیم و در انتها نیز با فراخوانی تابع complete\_ins\_fields فیلدهای مربوط به هر سهم را تکمیل می‌کنیم

* + 1. تابع get\_indexes

این تابع به منظوردریافت اطلاعات مربوط به شاخص‌ها تعبیه شده است؛ در این تابع، ابتدا کد یکتای مربوط به هر شاخص از فایل indexes.csv خوانده می‌شود و سپس با استفاده از URL با فرمت [http://cdn.tsetmc.com/api/Index/GetIndexB2History/{Index\_id}](http://cdn.tsetmc.com/api/Index/GetIndexB2History/%7bIndex_id%7d) اطلاعات مربوط به هر Index خوانده شده و در فیلد مربوط به آن Index در Object متناظر با تاریخ داده‌ی خوانده شده ذخیره می‌شود.

* + 1. تابع get\_day\_trade\_summaries

این تابع به منظور دریافت اطلاعات قیمتی یک سهم در یک تاریخ خاص تعبیه شده است؛ در این تابع با داشتن کد یکتای مربوط به یک سهم و با استفاده از URL با فرمت [http://cdn.tsetmc.com/api/ClosingPrice/GetClosingPriceDaily/{Instrument\_id}/{Date}](http://cdn.tsetmc.com/api/ClosingPrice/GetClosingPriceDaily/%7bInstrument_id%7d/%7bDate%7d) اطلاعات مربوط به معاملات یک روز خاص از نماد مورد نظر را دریافت کرده و متناظر با آن یک Object از نوع DayTradeSummary می‌سازیم.

* + 1. تابع get\_client\_trade\_infos

این تابع به منظور دریافت اطلاعات مربوط به معاملات مشتریان حقیقی و حقوقی یک سهم در یک تاریخ خاص تعبیه شده است؛ در این تابع با داشتن کد یکتای مربوط به یک سهم و با استفاده از URL با فرمت [http://cdn.tsetmc.com/api/ClientType/GetClientTypeHistory/{Instrument\_id}/{Date}](http://cdn.tsetmc.com/api/ClientType/GetClientTypeHistory/%7bInstrument_id%7d/%7bDate%7d) اطلاعات مربوط به معاملات یک روز خاص از نماد مورد نظر را دریافت کرده و متناظر با آن یک Object از نوع ClientTradeInfo می‌سازیم.

* + 1. تابع get\_change\_capitals

تابع به منظور دریافت اطلاعات مربوط به افزایش‌ سرمایه‌های تمامی سهم‌های بازار بورس تعبیه شده است؛ در این تابع با استفاده از URL با فرمت [http://tsetmc.com/Loader.aspxPartree=151310&Flow={flow}](http://tsetmc.com/Loader.aspxPartree=151310&Flow=%7bflow%7d) به اطلاعات مربوط به افزایش سرمایه تمامی سهم‌ها دسترسی خواهیم داشت؛ بنابراین به ازای هر افزایش سرمایه یک Object از نوع ChangeCapital می‌سازیم و مقادیر آن را برای سهم مورد نظر Set می‌کنیم.

* + 1. تابع calculate\_balanced\_price

این تابع به منظور محاسبه قیمت تعدیل‌شده سهم‌ها با در نظر گرفتن افزایش سرمایه‌های اتفاق افتاده برای آن‌ها تعبیه شده است؛ در این تابع برای هر سهم، ChangeCapitalهای متناظر را یافته و در بازه زمانی معتبر آن افزایش سرمایه، قیمت را با در نظر گرفتن سرمایه جدید تعدیل کرده و مقادیر مربوطه در Objectهای DayTradeSummaryهای بازه تاریخی موردنظر را به روزرسانی می‌کنیم.

* + 1. تابع create\_daily\_data

این تابع به منظور تجمیع و آماده‌سازی دیتاست نهایی تعبیه شده است؛ در این تابع با تجمیع داده‌های درون جداول DayTradeSummary، ClientTradeInfo و Indexes دیتاست نهایی قابل استفاده در سامانه تولید می‌شود.

* + 1. تابع unique\_crawl

این تابع به منظور واکشی اطلاعات (اطلاعات قیمتی و اطلاعات معاملات سهامداران) یک سهم خاص در یک بازه زمانی خاص تعبیه شده است؛ در این تابع به ازای هر روز در بازه زمانی تعریف شده در ورودی تابع، یکبار تابع get\_client\_trade\_infos برای دریافت اطلاعات سهامداران و یکبار تابع get\_day\_trade\_summaries برای دریافت اطلاعات قیمتی سهم در آن روز صدا زده می‌شوند.

* + 1. تابع setup

این تابع به منظور انجام تنظیمات اولیه (دریافت اطلاعات کلی سهم‌ها و شاخص‌ها و افزایش سرمایه‌ها) تعبیه شده است؛ در این تابع با فراخوانی توابع get\_indexes، get\_instruments و get\_change\_capitals اطلاعات مربوط به شاخص‌ها، سهم‌ها و افزایش سرمایه‌ها را واکشی و در پایگاه داده ذخیره می‌نماییم تا بتوانیم در ادامه عملیات خزش را آغاز کنیم.

* + 1. تابع crawl

این تابع به منظور واکشی اطلاعات معاملاتی تمامی سهم‌ها در یک بازه تاریخی خاص تعبیه شده است؛ در این تابع به ازای هر سهم، تابع unique\_crawl فراخوانده می‌شود تا اطلاعات معاملاتی آن سهم واکشی شود و سپس با فراخوانی تابع calculate\_balanced\_price قیمت‌های تعدیل‌شده سهم را نیز به پایگاه داده می‌افزاییم. در انتها نیز برای آماده‌سازی دیتاست، تابع create\_daily\_data را فرا می‌خوانیم.

* + 1. تابع add\_new\_instrument

این تابع به منظور افزودن اطلاعات کلی و قیمتی یک سهم به دیتاست به صورت دستی تعبیه شده است؛ در این تابع ابتدا Object مربوط به سهم یعنی Instrument ساخته می‌شود، سپس اطلاعات بنیادی آن به وسیله ساخت یک Object از نوع Codal اضافه می‌شود و پس از تکمیل تمامی فیلدهای سهم، با فراخوانی تابع unique\_crawl اطلاعات قیمتی و معاملاتی سهم را واکشی می‌کنیم و با استفاده از توابع get\_change\_capitals و calculate\_balanced\_price اطلاعات قیمتی سهم را تعدیل می‌کنیم.

* 1. خلاصه و جمع‌بندی

در این فصل به تشریح مدل داده‌ای پایگاه داده مربوط به خزنده و نیز اجزای اصلی ماژول خزنده در پروژه پرداختیم و نحوه کارکرد هر یک را به تفصیل توضیح دادیم.

1. راه‌اندازی و اجرای خزنده

در این فصل در مورد نحوه‌ی راه‌اندازی و اجرای خزنده بر روی localhost، استفاده از آن بر روی سرور میزبان و استفاده از سامانه مستقر شده توضیح داده خواهد شد.

* 1. مقدمه

پس از ارائه جزنیات فنی پروژه در فصل‌های قبلی، در این فصل قصد داریم تا مرحله به مرحله اجرای خزنده را شرح دهیم و در نهایت آن را راه‌اندازی نماییم؛ همچنین با توجه به اینکه این خزنده بر روی یک سرور لینوکس راه‌اندازی شده است، در این فصل نحوه کار با آن بر روی سرور و نیز استفاده از وبسایت سامانه پویش اطلاعات سهام را نیز توضیح خواهیم داد.

* 1. راه‌اندازی خزنده بر روی Localhost

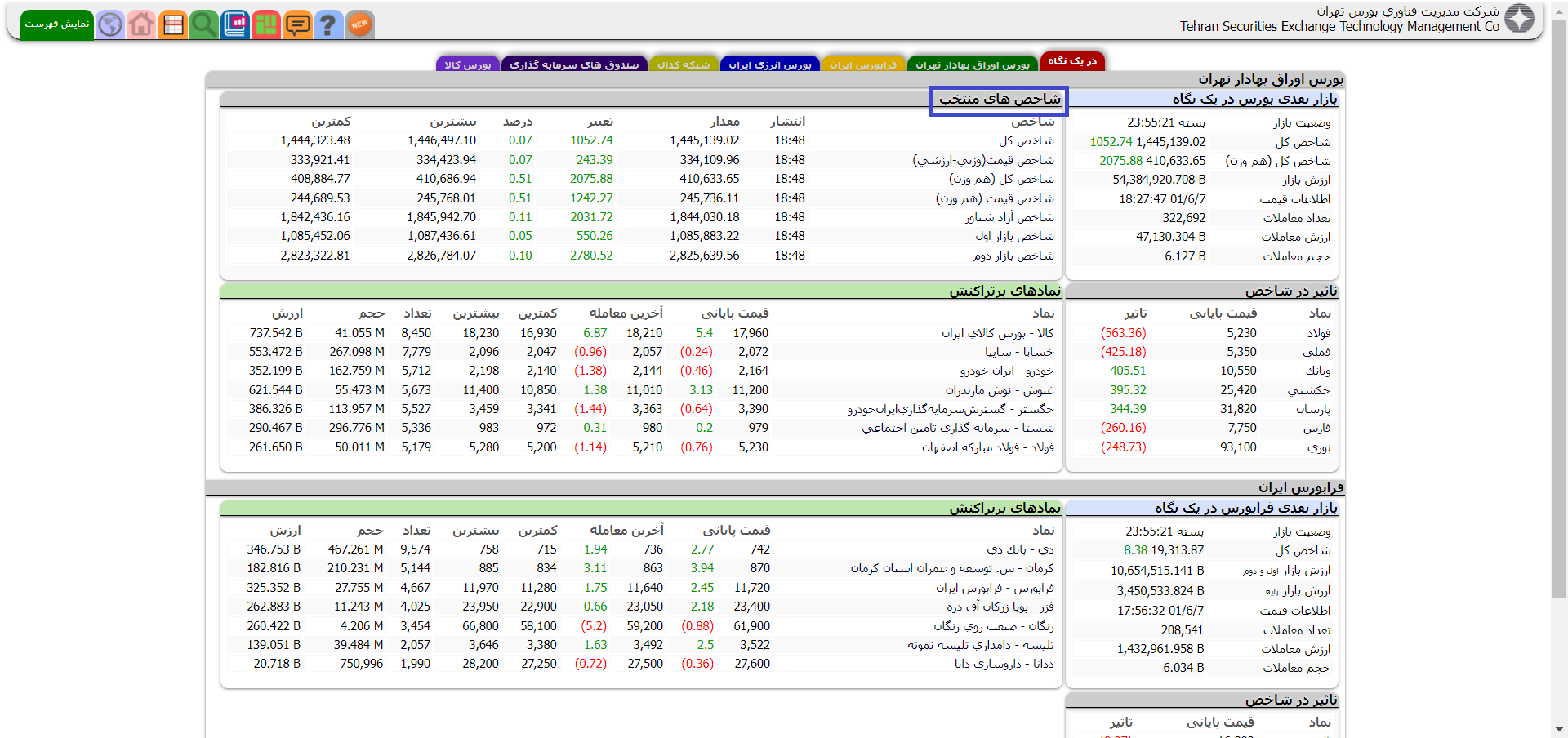
در این بخش به تشریح مراحل اجرای خزنده بر روی localhost خواهیم پرداخت.

### 4-2-1- آماده‌سازی فایل‌های Setup اولیه

در ابتدا لازم است تا کدهای یکتای مربوط به شاخص‌ها و نمادها را در اختیار پروژه قراردهیم تا از آن‌ها برای تکمیل فرمت URLهای مربوط به وبسایت سازمان بورس استفاده نماید؛ بدین منظور لازم است تا دو فایل با نام‌های indexes.csv و ins.csv بسازیم و در فولدر اصلی پروژه قرار دهیم (لازم به ذکر است که این دو فایل اکنون در فایل مربوط به پروژه قرار دارند)؛ حال به ترتیب هر یک از دو فایل را باید تکمیل نماییم؛

#### 4-2-1-1- تکمیل فایل indexes.csv

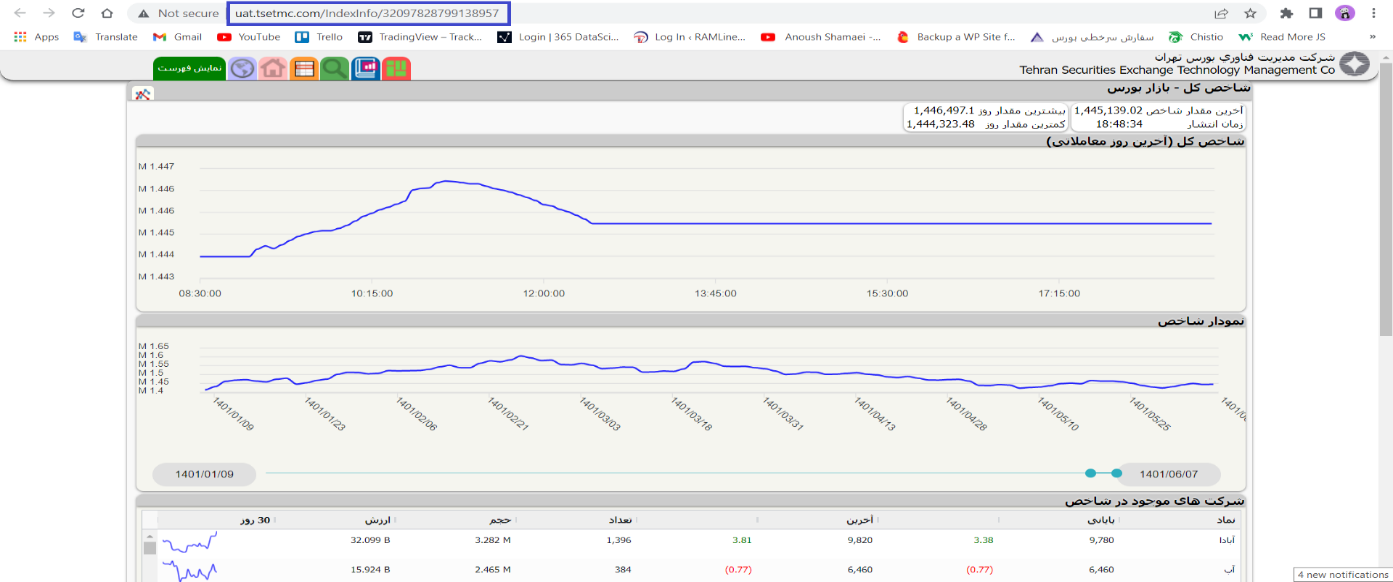
در این بخش باید کدهای یکتای مربوط به هر شاخص را در این فایل Copy نماییم؛ بدین منظور مطابق (شکل 12-4) از صفحه اصلیسایت tsetms.com شاخص‌های منتخب را انتخاب نموده و در لیست نمایش داده شده (شکل 13-4) شاخصی که می‌خواهیم اطلاعات آن را اضافه نماییم را انتخاب می‌کنیم و وارد صفحه آن می‌شویم (شکل 14-4)؛ سپس از انتهای URL آن کد یکتا را کپی و در فایل Paste می‌نماییم. تصویر نهایی فایل indexes.csv در (شکل 15-4) قابل مشاهده است:



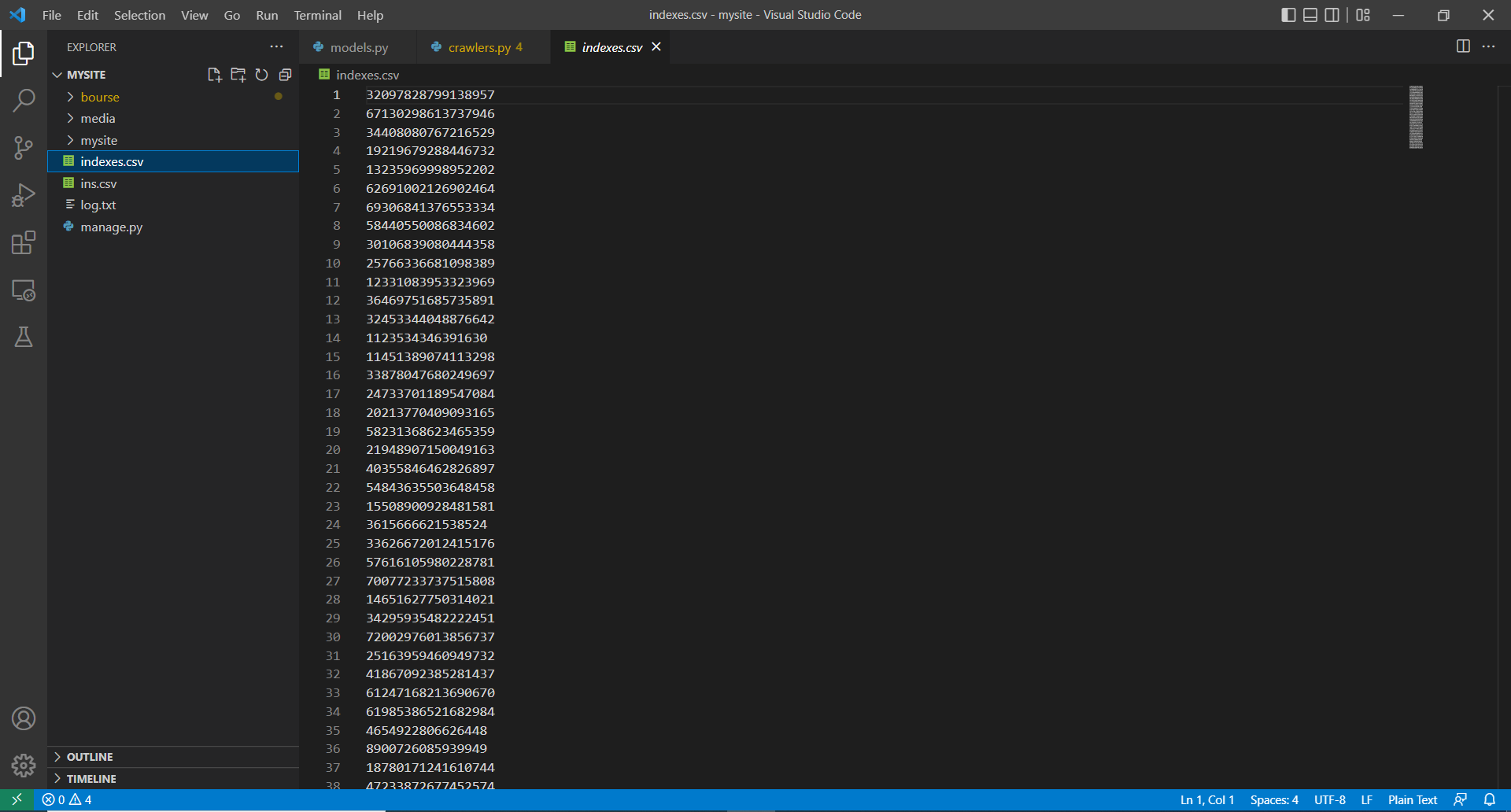
شکل 13-4) انتخاب شاخص‌های منتخب



شکل 14-4) لیست تمامی شاخص‌ها



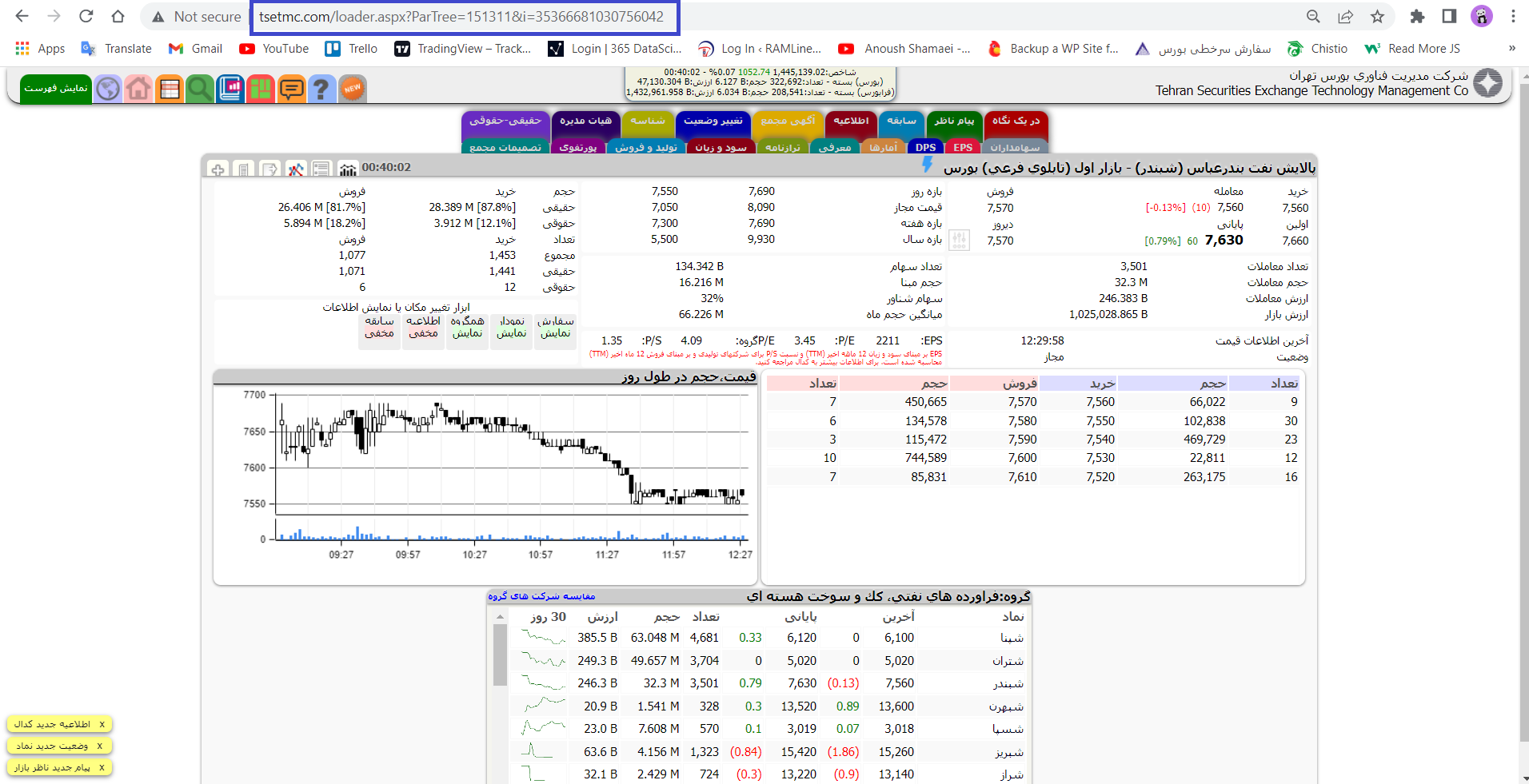
شکل 15-4) صفحه مربوط به شاخص‌ کل و کد یکتای آن



شکل 16-4) تصویر فایل indexes.csv

#### 4-2-1-2- تکمیل فایل ins.csv

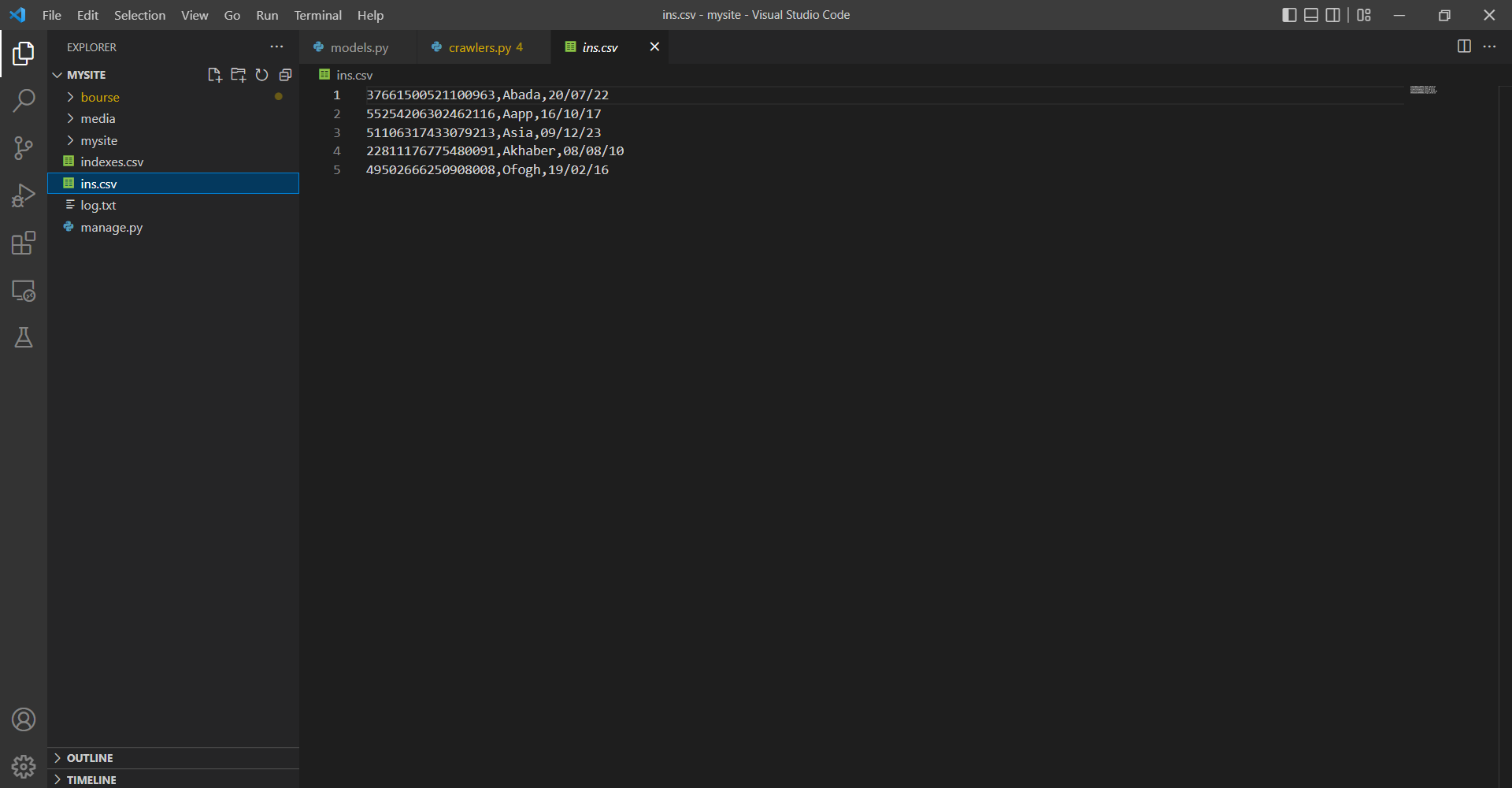
در این بخش باید کدهای یکتای مربوط به نمادها (و به صورت اختیاری نام انگلیسی نماد و تاریخ عرضه آن) را در این فایل Copy نماییم؛ کد یکتای هر نماد از انتهای URL صفحه اصلی مربوط به آن قابل استخراج است (شکل 16-4)، نام انگلیسی نماد توسط خود شما انتخاب می‌شود و تاریخ عرضه آن در Tab با نام «سابقه» در صفحه اصلی مربوط به سهم (شکل 17-4) قابل دستیابی است؛ در این Tab به آخرین معامله انجام شده می‌رویم و تاریخ آن را به فایل می‌افزاییم. تصویر نهایی فایل ins.csv در (شکل 18-4) قابل مشاهده است؛ مجددا تاکید می‌کنم که تنها وارد کردن کد یکتای هر نماد در فایل اجباری است و وارد نکردن نام انگلیسی و تاریخ عرضه مشکلی ایجاد نمی‌نماید:



شکل 17-4) صفحه اصلی مربوط به نماد



شکل 18-4) صفحه مربوط به سابقه معاملاتی نماد



شکل 19-4) تصویر فایل نهایی ins.sv

### 4-2-2- ساخت پایگاه داده مربوط به پروژه در mySQL

پس از آنکه mySQL را برروی سیستم خود نصب نمودید؛ لازم است تا پایگاه داده مربوط به پروژه را در آن بسازید. بدین منظور ابتدا با استفاده از دستور زیر وارد داشبورد mySQL می‌شویم:

sudo mysql -u root

در صورتی که اولین بار است وارد داشبورد می‌شوید، لازم است تا یک user بسازید, این کار با اجرای دستور زیر قابل انجام است:

CREATE USER 'db\_user'@'localhost' IDENTIFIED WITH mysql\_native\_password BY 'P@s$w0rd123!';

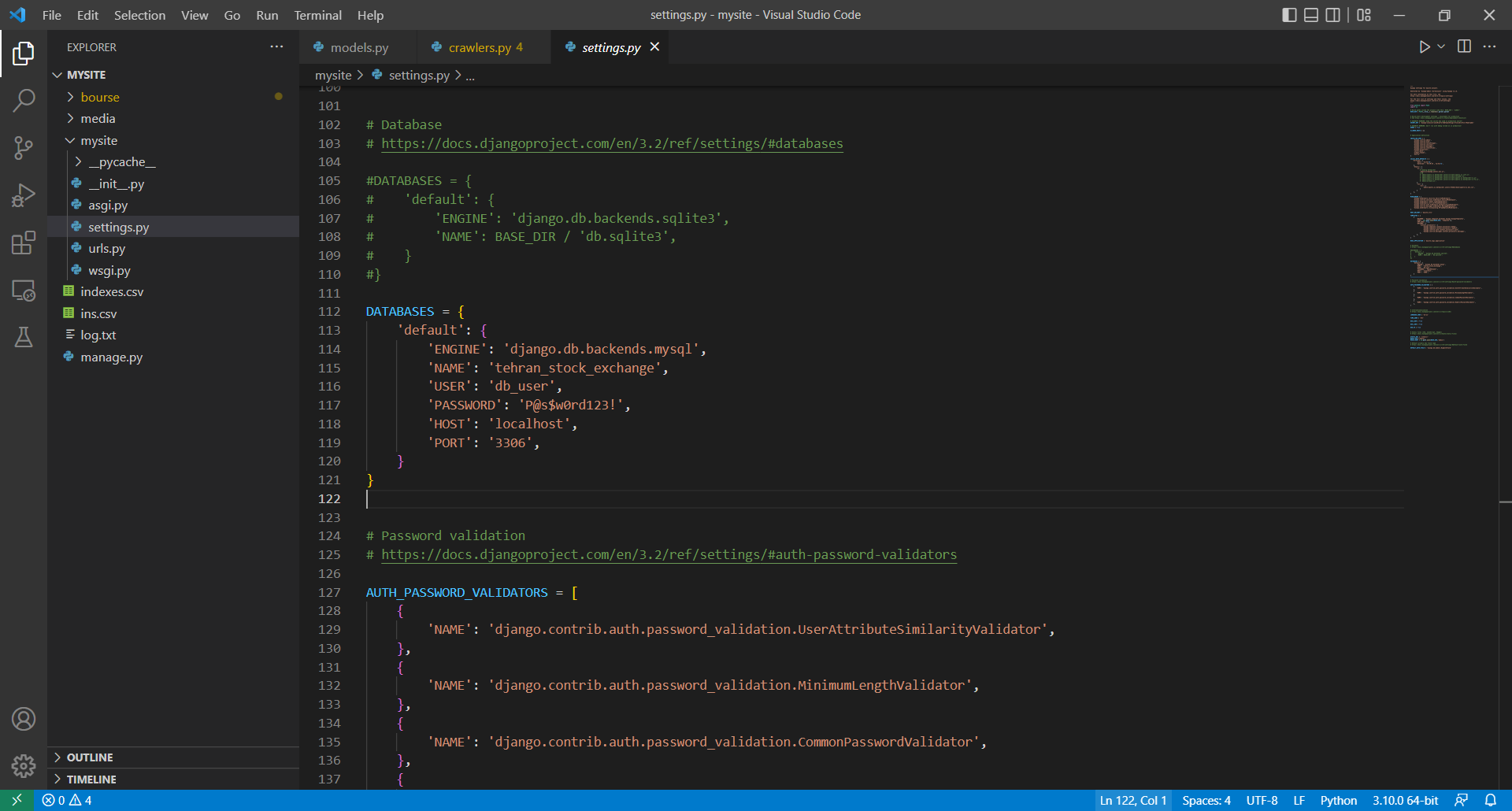
سپس پایگاه داده مربوط به پروژه خود را با اجرای دستور زیر بسازید:

CREATE DATABASE tehran\_stock\_exchange;

لازم است تا permissionهای لازم را با استفاده از دستور زیر برای پایگاه داده خود set کنیم:

GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, INDEX, DROP, ALTER, REFERENCES, CREATE TEMPORARY TABLES, LOCK TABLES ON tehran\_stock\_exchange.\* TO 'db\_user'@'localhost';

در انتها لازم است تا ویژگی‌های مربوط به پایگاه داده را به تنظیمات پروژه Django خود بیفزایید؛ به این منظور در پوشه پروژه وارد فایل mysite/settings.py شده و در قسمت DATABASES اطلاعات مربوط به پایگاه داده را بیفزایید(شکل 19-4):



شکل 20-4) تنظیمات پایگاه داده در settings.py

### 4-2-3- اجرای خزنده

برای اجرای پروژه Django مربوط به خزنده ابتدا لازم است تا migrationهای مربوط به پایگاه داده را انجام دهید؛ بدین منظور دو دستور زیر را در ترمینال اجرا نمایید:

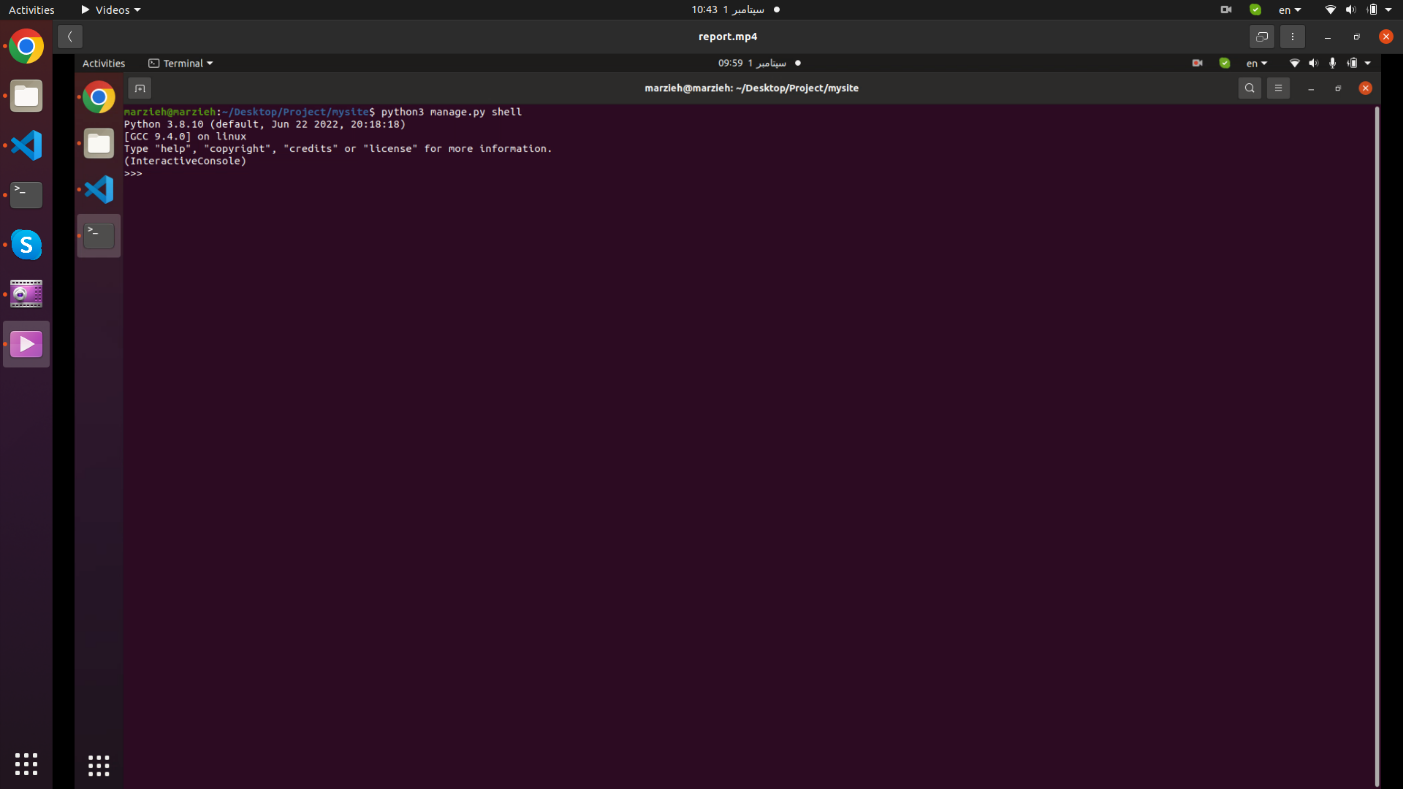
python3 manage.py makemigrations

python3 manage.py migrate

سپس برای اجرای کدهای مربوط به خزنده، باید وارد shell شوید؛ این کار را اجرای دستور زیر انجام می‌دهید:

python3 manage.py shell

شکل 20-4 تصویر shell مربوط به Django را نشان می‌دهد:



شکل 21-4) تصویر Shell مربوط به Django

حال همه چیز برای اجزا خزنده و پرکردن جداول پایگاه داده آماده است؛ ابتدا لازم است تا توابع مورد نیاز برای اجرای خزنده را import نمایید؛ بدین منظور از دستور زیر استفاده می‌کنید:

from bourse.services.crawlers import \*

سپس باید داده‌های مربوط به شاخص‌ها، اطلاعات کلی نمادها و افزایش سرمایه‌ها را واکشی کنید؛ تمامی این کارها با فراخوانی تابع setup قابل انجام است:

setup()

حال می‌توانید خزنده داده‌های معاملاتی را اجرا نمایید؛ بدین منظور باید تابع crawl را به همراه بازه زمانی واکشی فراخوانی کنید. لازم به ذکر است که وارد کردن انتهای بازه واکشی اجباری نیست و اگر آن را وارد نکنید، به صورت خودکار از ابتدای بازه تا تاریخ همان روز را crawl می‌کند؛ برای مثال اگر می‌خواهید داده‌ها را از تاریخ 2020/08/06 تا امروز crawl کنید، باید دستور زیر را وارد نمایید:

crawl(date(2020, 8, 6))

با اجرای مراحل بالا، جداول پایگاه داده پر شده و سامانه آماده استفاده است. 😊

### 4-2-4- اجرای سامانه پویش داده بر روی localhost

از shell با استفاده از دستور exit() خارج شوید و در فولدر مربوط به پروژه دستور زیر را وارد نمایید:

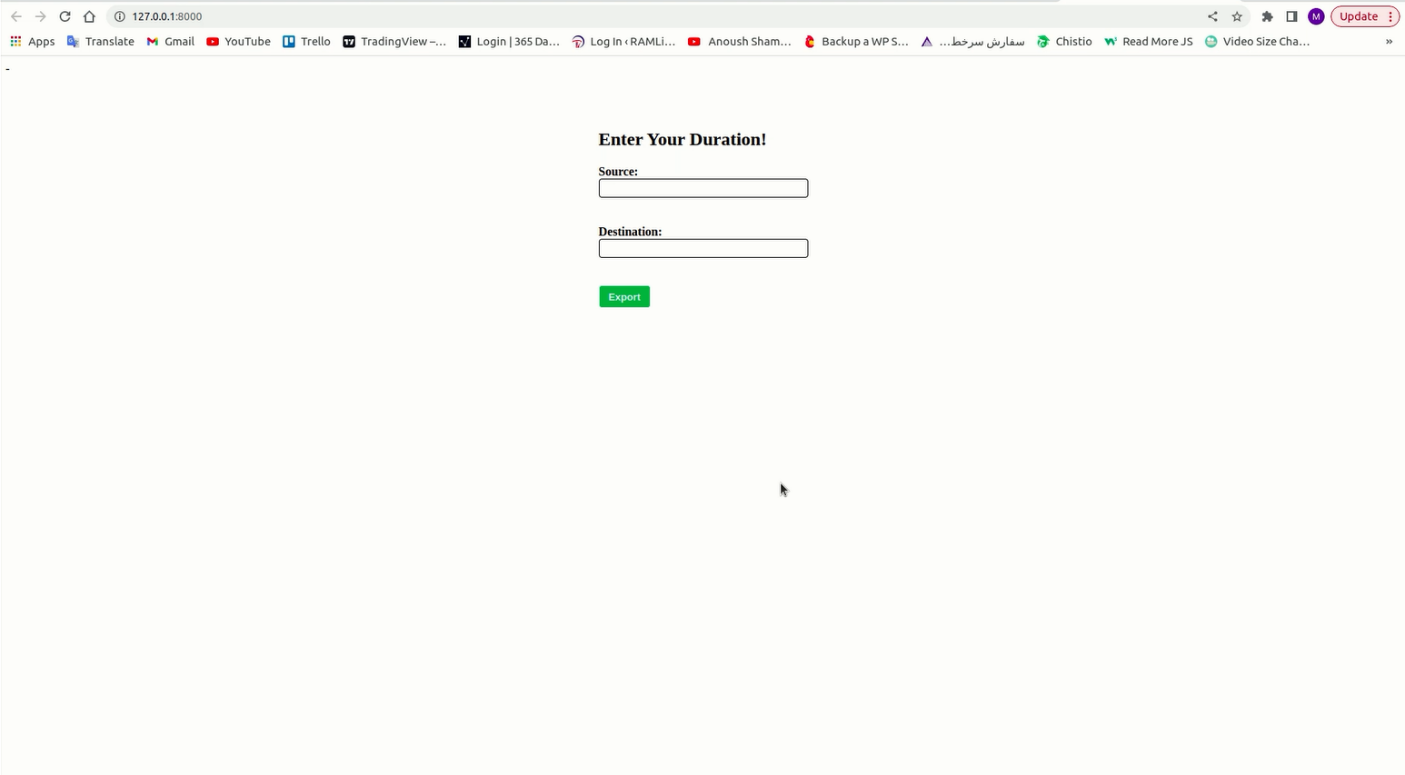
python3 manage.py runserver

در صورتی که تمامی مراحل فوق را به درستی انجام داده باشید، خروجی دستور به صورت زیر خواهد بود:



شکل 22-4) خروجی اجرای دستور runserver

همانطور که در (شکل 21-4) قابل مشاهده است، سامانه بر روی localhost با آدرس <http://127.0.0.1:8000/> قابل دسترسی است؛ با باز کردن این لینک بر روی browser خروجی زیر را مشاهده خواهید کرد:



شکل 23-4) سامانه پویش و پردازش داده بورس تهران

نحوه کار با سامانه به اینگونه است، که اگر فیلدهای Source و Destination را خالی گذاشته و تنها بر روی Export کلیک نمایید،کل داده‌های معاملاتی موجود در سامانه را در قالب .csv خروجی می‌دهد و اگر فیلدهای Source و Destination را پر نمایید، داده‌های معاملاتی موجود در بازه زمانی مورد نظر شما را خروجی می‌دهد.

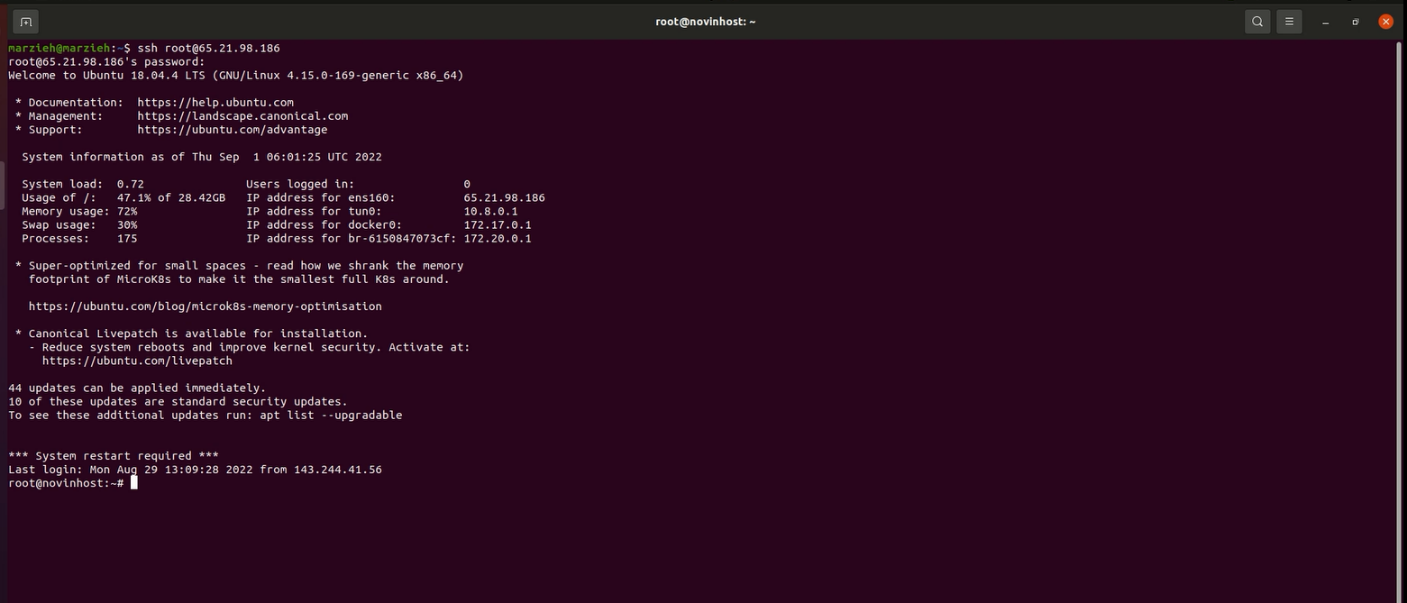
* 1. کار با خزنده بر روی سرور Linux

همانطور که می‌دانید، هدف از اجرای این پروژه راه‌اندازی یک سامانه (وبسایت) جهت دریافت اطلاعات معاملاتی بازار بورس تهران بود؛ بنابراین لازم بود تا پروژه آماده شده را بر روی یک server واقعی deploy نماییم؛ بدین منظور ما از یک سرور لینوکسی با ip برابر با 65.21.98.186 استفاده نمودیم. حال برای دسترسی به پروژه بر روی این سرور و ویرایش یا کار با آن باید به شرح زیر عمل نماییم:

در ابتدا باید وارد محیط سرور شویم؛ این کار را با اجرای دستور زیر در ترمینال و سپس وارد کردن پسورد مربوط به سرور انجام می‌دهیم:

ssh [root@65.21.98.186](mailto:root@65.21.98.186)

خروجی دستور باید به شکل زیر باشد:



شکل 24-4) ترمینال سرور لینوکسی

حال ما وارد ترمینال مربوط به سرور لینوکسی شده‌ایم و دقیقا مطابق با ترمینال سیستم‌ خود می‌توان با آن کار نمود. با توجه به اینکه پروژه را برروی apache، deploy کرده‌ایم، فایل‌های مربوط به پروژه در فولدر /var/www قرار داده شده است؛ بنابراین با اجرای دستور زیر می‌توان وارد فولدر پروژه شد:

cd /var/www/mysite

برای ویرایش یک فایل می‌توان از دستور nano استفاده نمود؛ برای مثال برای ویرایش فایل crawlers.py واقع در فولدر bourse/services از دستور زیر استفاده می‌نماییم:

nano bourse/services/crawlers.py

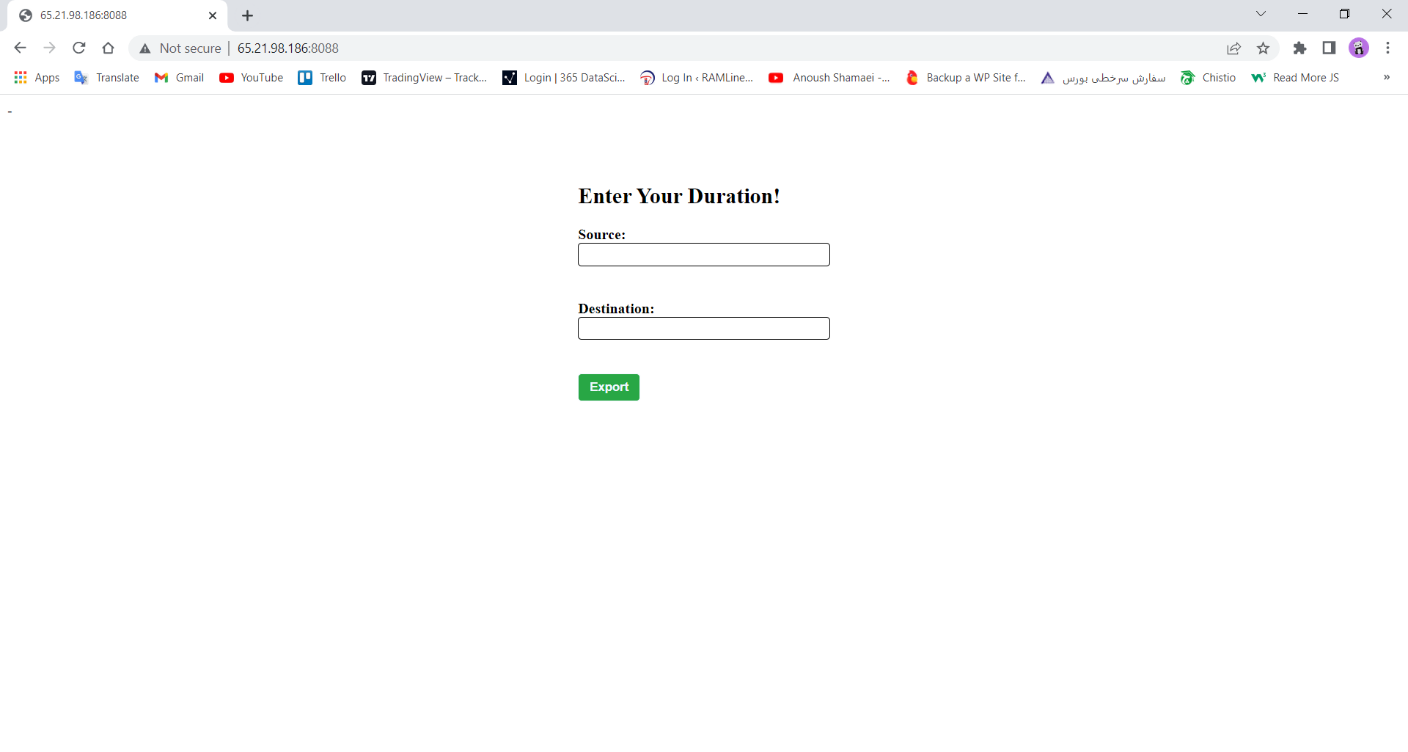
پس از اعمال تغییرات با استفاده از کلیدهای crrl+S تغییرات را ذخیره و با استفاده از کلیدهای ctrl+X از فایل خارج شده و به محیط ترمینال بر می‌گردیم. سپس برای آنکه تغییرات اعمال شده بر روی سامانه اجرا شود باید از دستور زیر استفاده کنیم:

service apache2 reload

سایر موارد کار با سرور (اعم از دستورات مربوط به migration، shell و ...) همانند کار با پروژه بر روی سیستم است که در بخش قبل توضیح داده شد.

* 1. کار با سامانه پویش و آماده‌سازی داده‌های بورس تهران

وبسایت سامانه پویش داده‌های بورس تهران، بر روی آدرس <http://65.21.98.186:8088/> قابل مشاهده است؛ تصویر این وبسایت را در ادامه ملاحظه می‌کنید:



شکل 25-4) تصویر وبسایت سامانه پویش و آماده‌سازی داده‌های بورس تهران

نحوه کار با سامانه به اینگونه است، که اگر فیلدهای Source و Destination را خالی گذاشته و تنها بر روی Export کلیک نمایید،کل داده‌های معاملاتی موجود در سامانه را در قالب .csv خروجی می‌دهد و اگر فیلدهای Source و Destination را پر نمایید، داده‌های معاملاتی موجود در بازه زمانی مورد نظر را خروجی می‌دهد.

* 1. خلاصه و جمع‌بندی

در این فصل به نحوه راه‌اندازی خزنده و کار با پروژه و اجرای آن بر روی localhost پرداختیم، سپس کار با پروژه بر روی یک سرور لینوکسی واقعی را بررسی کردیم و در آخر نیز با وبسایت مربوط به سامانه کار کردیم.

1. جمع‌بندی، نتيجه‌گيری و پيشنهادها

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

* 1. جمع‌بندی

در اين تحقيق در گام نخست ما با ساختار و نحوه چینش و قرارگیری اطلاعات در وبسایت سازما بورس تهران آشنا شدیم. سپس با مطالعه ‌و استفاده از مدل‌ داده‌ای و پیاده‌سازی خزنده‌ی قبلی، مدل داده‌ای را ویرایش و تکمیل نموده و خزنده جدید را طراحی و آماده کردیم؛ پس از پایان فرایند آماده‌ساز خزنده، پروژه را بر روی یک سرور لینوکسی Deploy کرده و خزنده را اجرا نمودیم تا داده‌ها را به صورت منظم استخراج نماید و همچنین رابط کاربری ساده و مناسبی پیاده‌سازی و مستقر کردیم تا با استفاده از آن افراد بتوانند داده‌های معاملاتی مربوط به نمادهای بازار بورس تهران را در بازه‌های تاریخی مورد نظر خود در قالب یک فایل .csv (به منظور سهولت در استفاده‌های بعدی) دریافت و دانلود نمایند.

* 1. نتيجه‌گيری
     1. دستاوردها

در نتیجه این پروژه، یک خزنده با قابلیت انعطاف در برابر تغییرات احتمالی وبسایت سازمان بورس و یک دیتاست کامل، دسته‌بندی شده و حاوی تمامی اطلاعات معاملاتی نمادها تهیه شد. همچنین یک وبسایت طراحی شد که با استفاده از آن به راحتی می‌توان به تمامی داده‌های واکشی‌شده توسط خزنده در هر بازه زمانی دلخواهی دسترسی پیدا کرد و آن‌ها را در یک قالب قابل استفاده در ابزارهای تحلیلی یعنی .csv دریافت نمود.

* + 1. پيشنهادها

مجموعه داده‌ای که در نتیجه این پروژه تهیه شده است و روزانه نیز به روز می‌شود، بسیار کامل و جامع است و قابلیت ویرایش، تجمیع و دسته‌بندی خوبی دارد؛ به همین علت برای استفاده به منظور ورودی تمامی ماژول‌های تحلیلی مانند اندیکاتورها، اسیلاتورها و مدل‌های یادگیری ماشین مناسب است. بنابراین پیشنهاد می‌شود تا از این داده‌ها برای توسعه ابزارهای تحلیلی (به خصوص در مدل‌های هوش مصنوعی) بهره گرفته شود تا از خروجی آن‌ها بتوان تحلیل و پیش‌بینی‎‌های با دقت بالایی از وضعیت قیمتی نمادها در بازار بورس تهران به دست آورد.

1. مراجع

در این فصل به معرفی مراجعی که در انجام این پروژه از آن‌ها استفاده نمودیم، می‌پردازیم.

مراجع

[1] Robert Layton, “Learning Data Mining with Python - Second Edition”, April 2013.

[2] Seppe vanden Broucke, Bart Baesens, “Practical Web Scraping for Data Science: Best Practices and Examples with Python”, April 2018.

[3] William S. Vincent, “Django for Professionals: Production websites with Python & Django “, July 2019.

1. Abstract [↑](#footnote-ref-1)
2. http://www.tsetmc.com [↑](#footnote-ref-2)
3. Crawl [↑](#footnote-ref-3)
4. http://www.tsetmc.com/Loader.aspx?ParTree=15 [↑](#footnote-ref-4)
5. http://www.tsetmc.com/Loader.aspx?ParTree=111C1417 [↑](#footnote-ref-5)
6. JavaScript Object Notation [↑](#footnote-ref-6)
7. Django [↑](#footnote-ref-7)
8. Object-Relational-Mapping [↑](#footnote-ref-8)
9. Field [↑](#footnote-ref-9)